

Board-Data Processing

VHDL Exercises

Exercise 1:

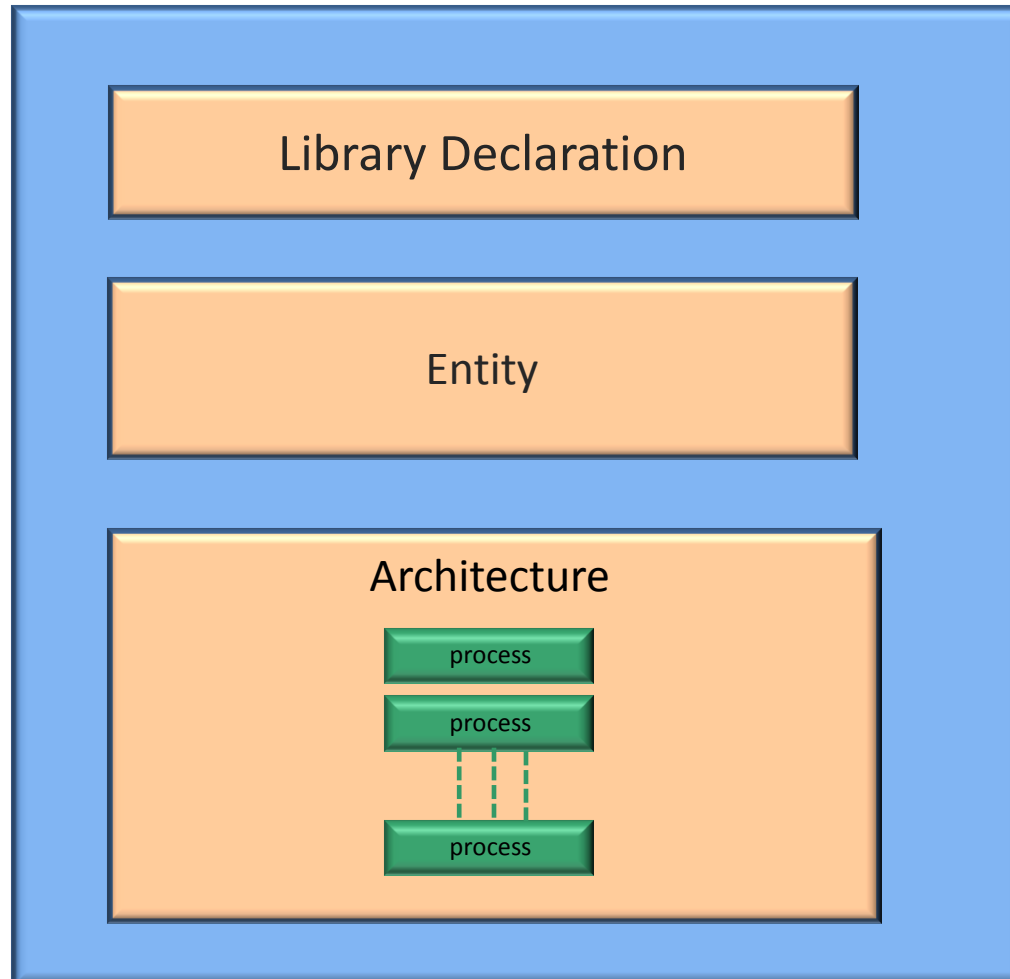
- Basics of VHDL Programming
- Stages of the Development process using FPGA's in Xilinx ISE.



Basics of VHDL

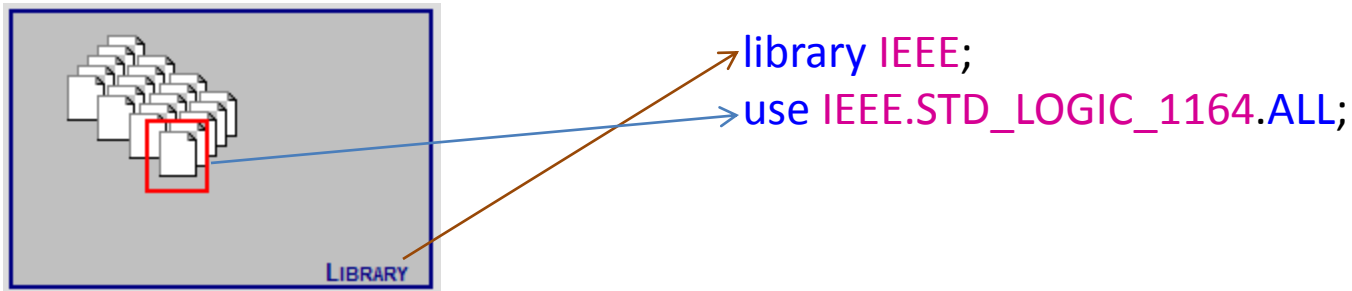
- VHDL (**Very High Speed IC Hardware description Language**) is one of the standard hardware description language used to design digital systems.

Structure of VHDL code



Structure of VHDL code

Library Declaration



STANDARD: Contains all basic declarations and definitions of language constructs and it is included in all VHDL specifications by default.

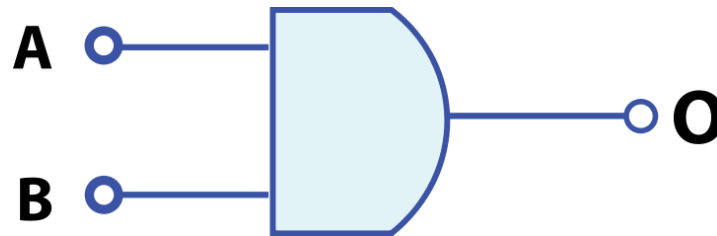
STD_LOGIC_1164: This package is not a part of the VHDL Standard but is a standard on its own; it contains the most often used language extensions.

TEXTIO: Contains declarations of basic operations on texts.

Structure of VHDL code

Entity

Defines the interface of the design



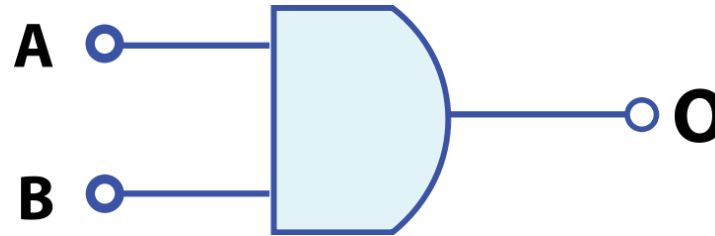
Interface

```
entity ANDGATE is  
  port (  
    A : in STD_LOGIC;  
    B : in STD_LOGIC;  
    O : out STD_LOGIC);  
end ANDGATE
```



Architecture

Describes the functionality of the design i.e transforming input data into output results.



Functionality

architecture Behavioral of ANDGATE is
begin
 O <= A and B;
end Behavioral;



Example1

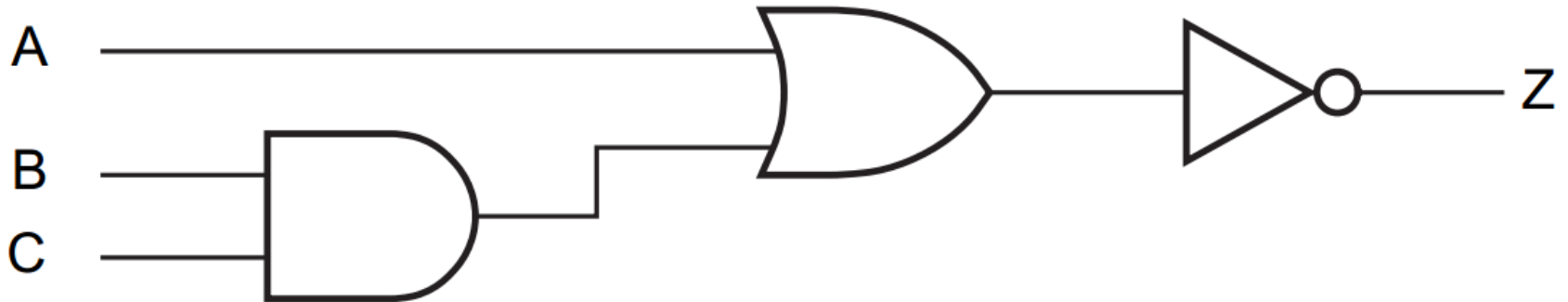
```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;
```

```
entity ANDGATE is  
  port (  
    A : in  STD_LOGIC;  
    B : in  STD_LOGIC;  
    O : out STD_LOGIC);  
end ANDGATE
```

```
architecture Behavioral of ANDGATE is  
begin  
    O <= A and B;  
end Behavioral;
```



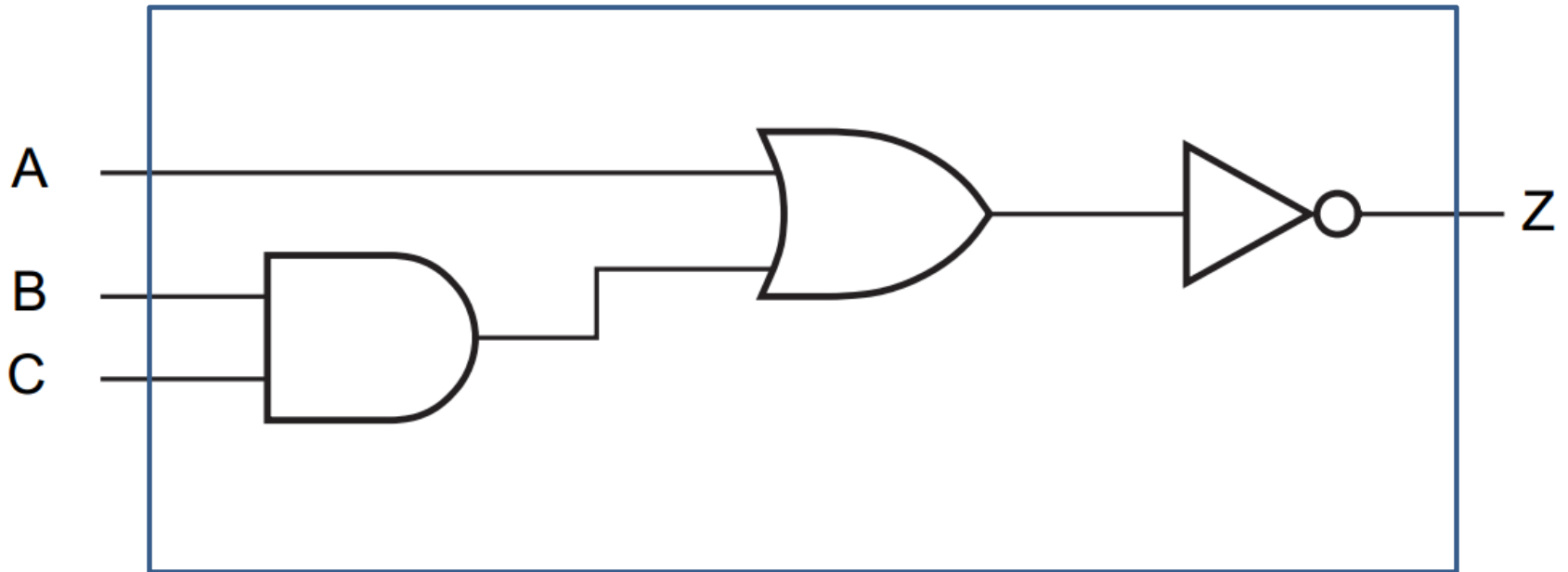
Signals



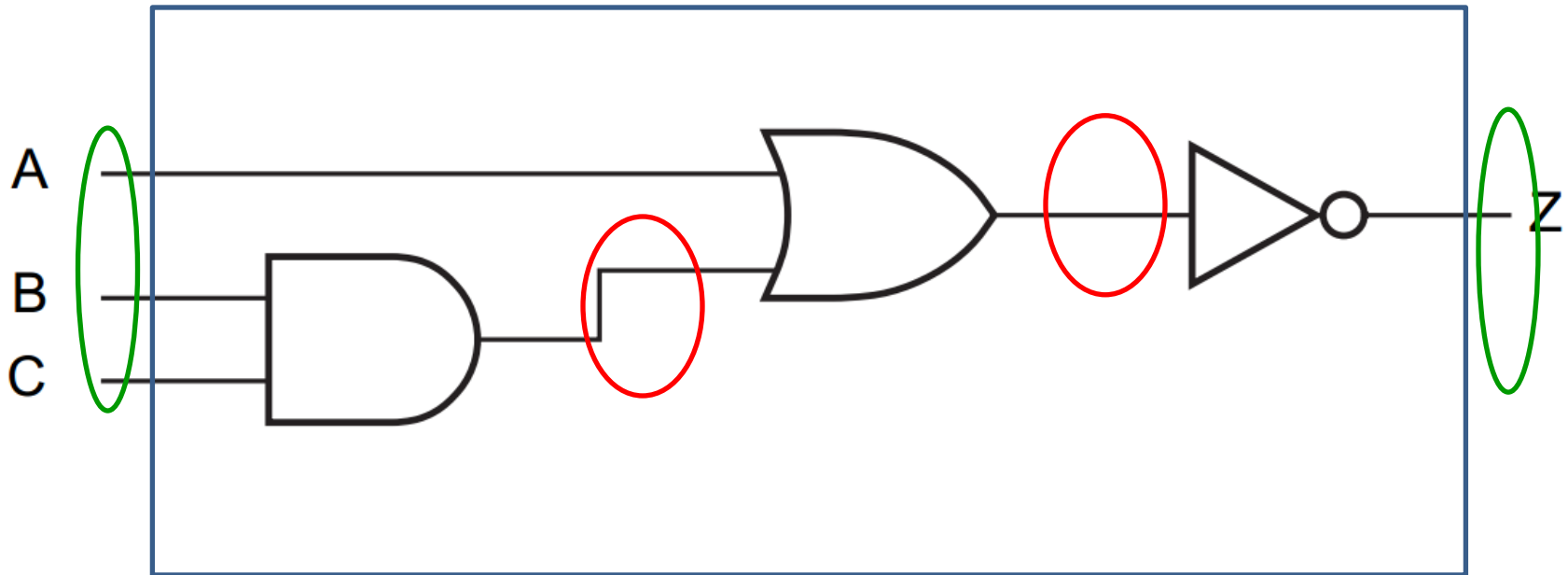
Signals



Signals



Signals



Signals

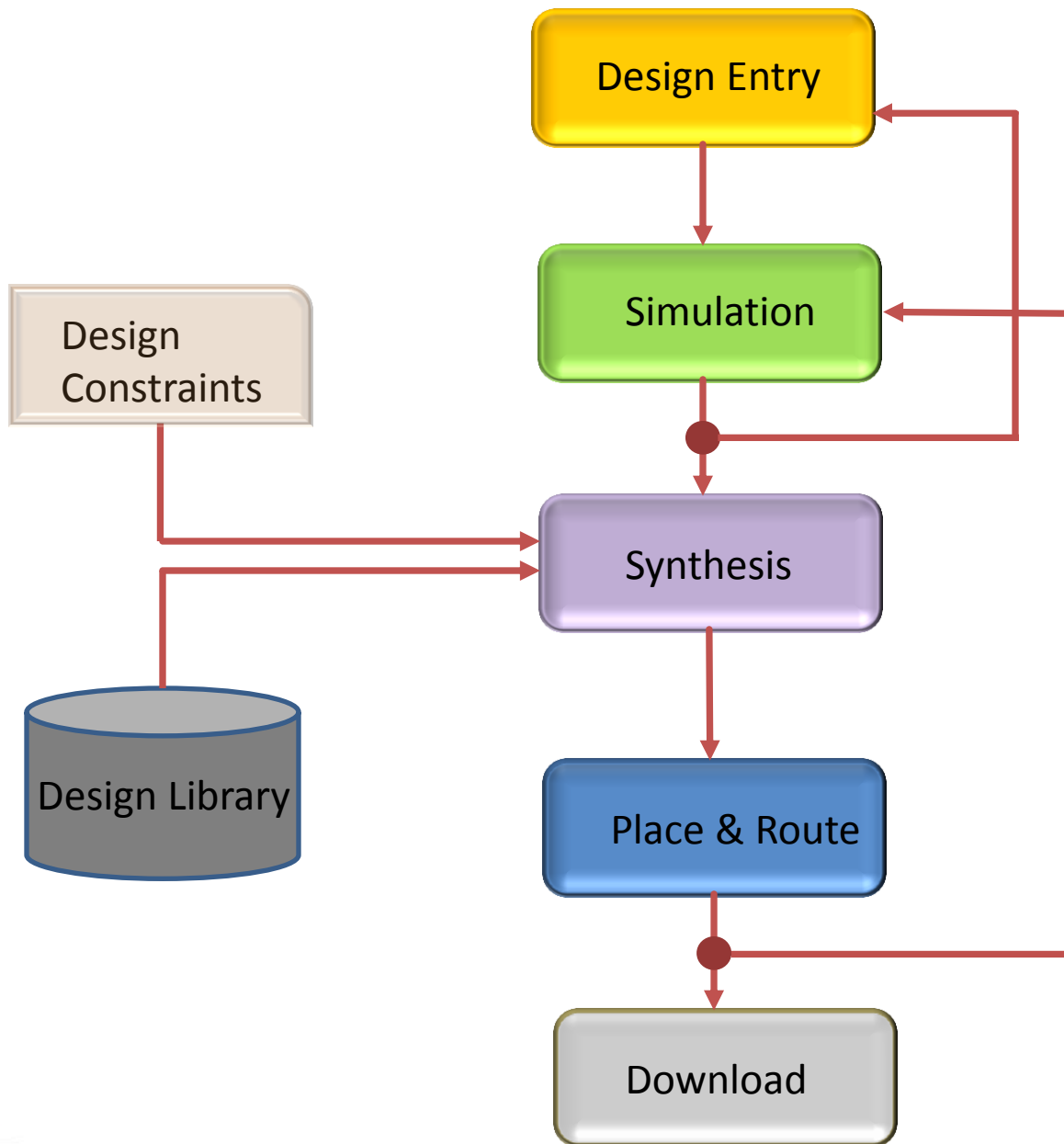
Signals represent wires or outputs of gates, FFs, etc. Ports (ins, outs, inout) in the entity are signals. Internal signals are often needed in complex models and are declared in the architecture description as follows:

```
architecture architecture_name of entity_name is  
    signal signal_name: type;  
    :  
    :  
    signal signal_name: type;  
begin  
    :  
    :  
end architecture architecture_name;
```



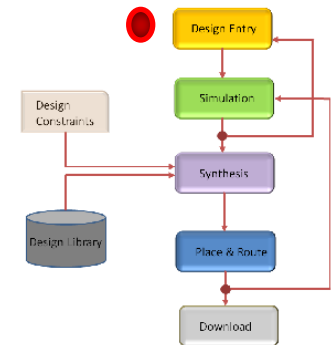
Stages of the Development of a Digital process in FPGA's





Design Entry

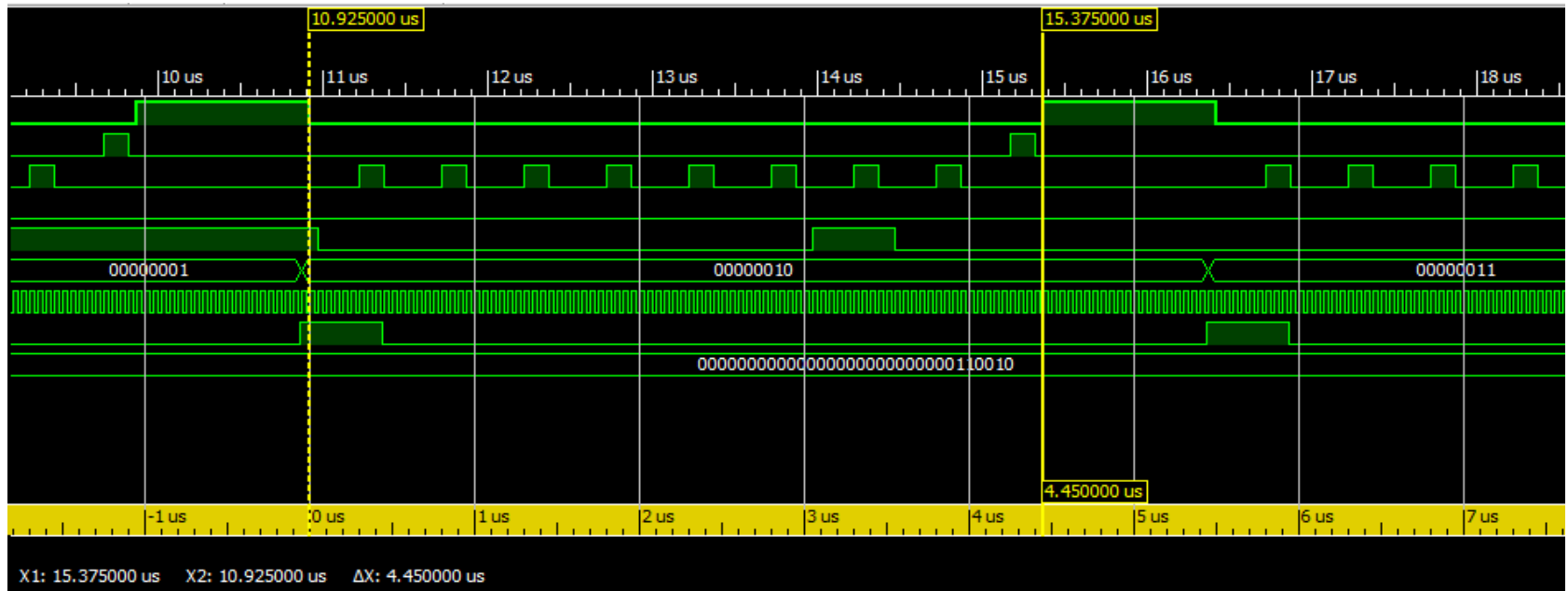
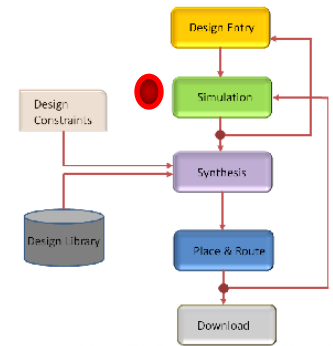
-> Writing the VHDL or VERILOG code



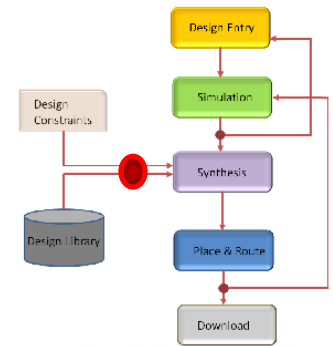
```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity timer is
7      Port ( clk : in  STD_LOGIC;
8            reset : in  STD_LOGIC;
9            interrupt: in  STD_LOGIC;
10           idle : inout STD_LOGIC;
11           sec : inout STD_LOGIC_VECTOR(4 downto 0));
12 end timer;
13 architecture Behavioral of timer is
14     signal cnt_int : std_logic_vector(4 downto 0);
15 begin
16     process (clk)
17     begin
18         if clk='1' and clk'event then
19             if (reset='1') then
20                 idle <= '0';
21                 cnt_int <= "00000";
22                 sec <= "00000";
23             end if;
24             if (interrupt = '0' and not(sec = "10000")) then
25                 cnt_int <= cnt_int + 1;
26                 if cnt_int="10010" then
27                     sec <= sec + 1;
28                     cnt_int <= "00000";
29                 end if;
30             end if;
31             if sec="10000" then
32                 idle <= '1';
33             else
34                 idle <= '0';
35             end if;
36         end if;
37     end process;
38 end Behavioral;
```

Simulation

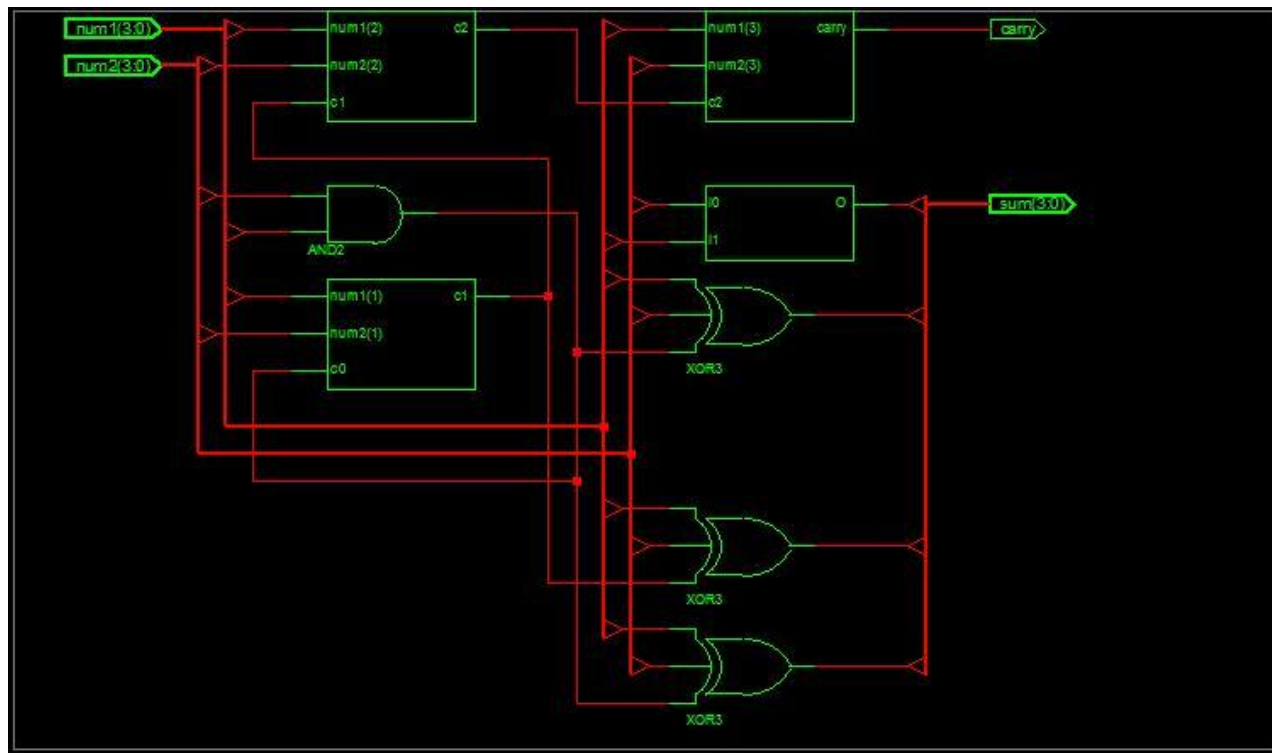
-> Identify Syntax and logic errors



Synthesis

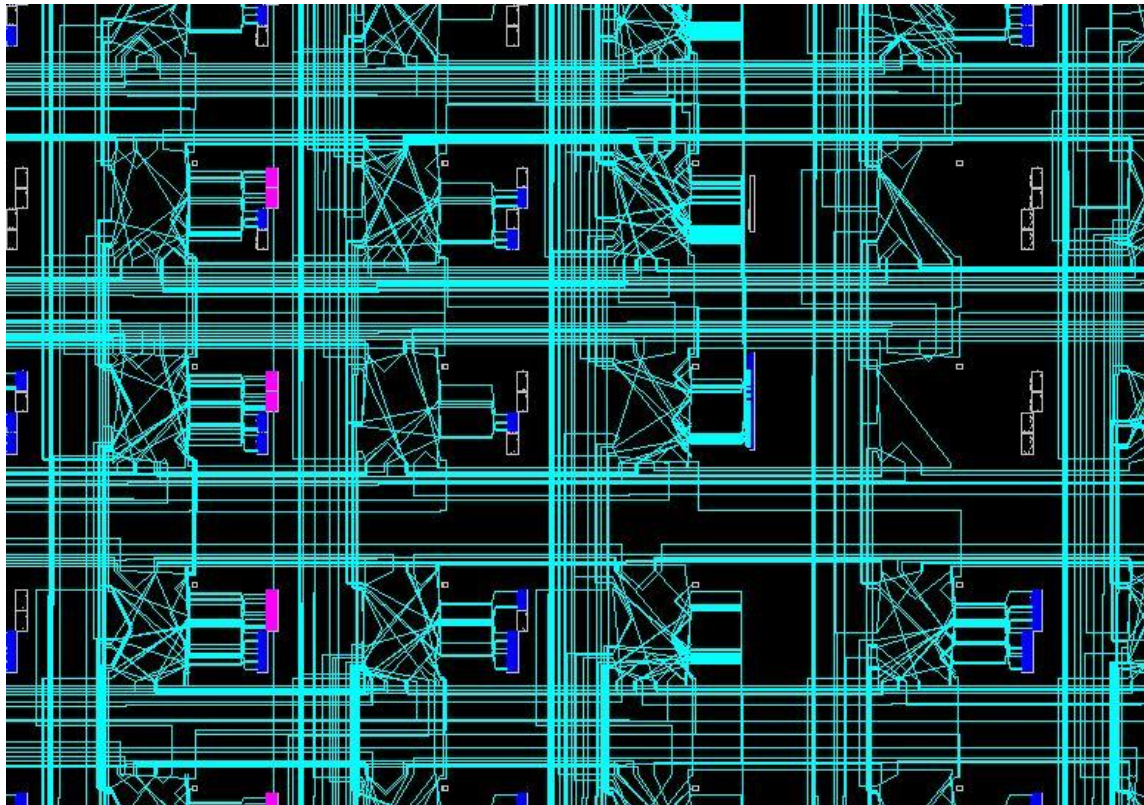
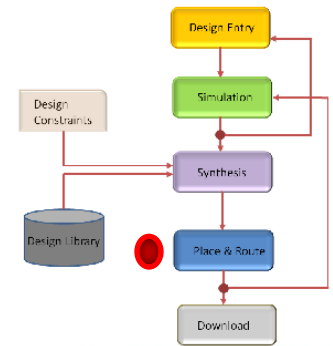


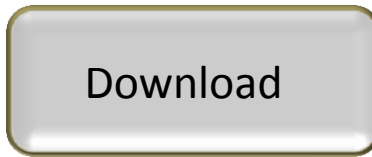
-> Generate Netlist



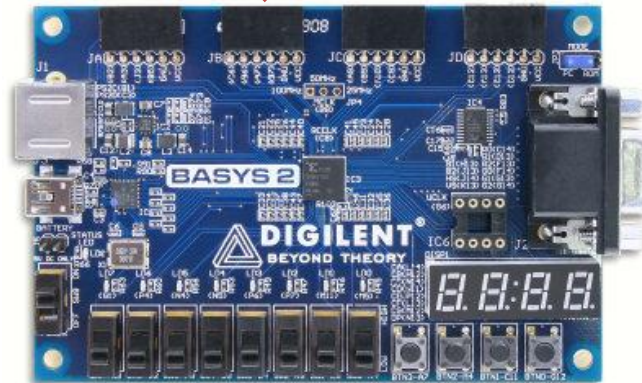
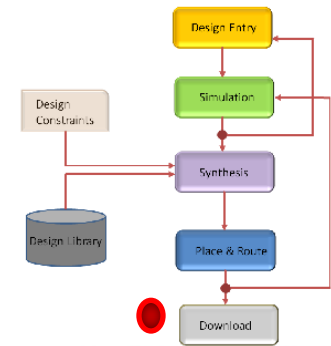
Place & Route

-> Map, Place & Route and generate the binary file

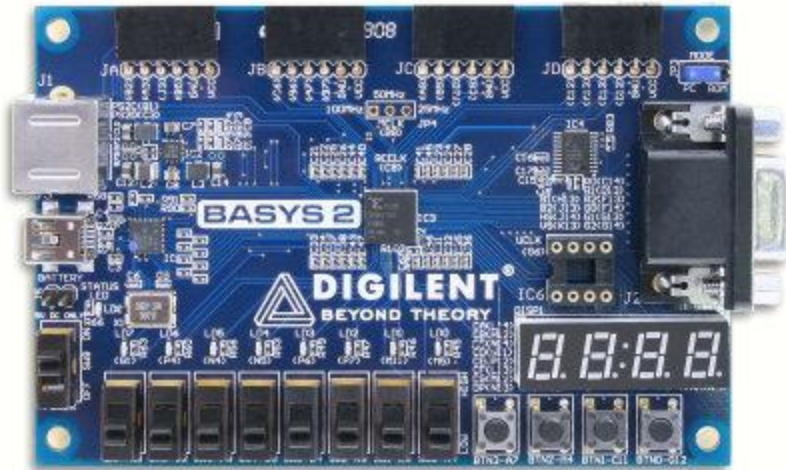




-> Flash the binary file on the FPGA development board



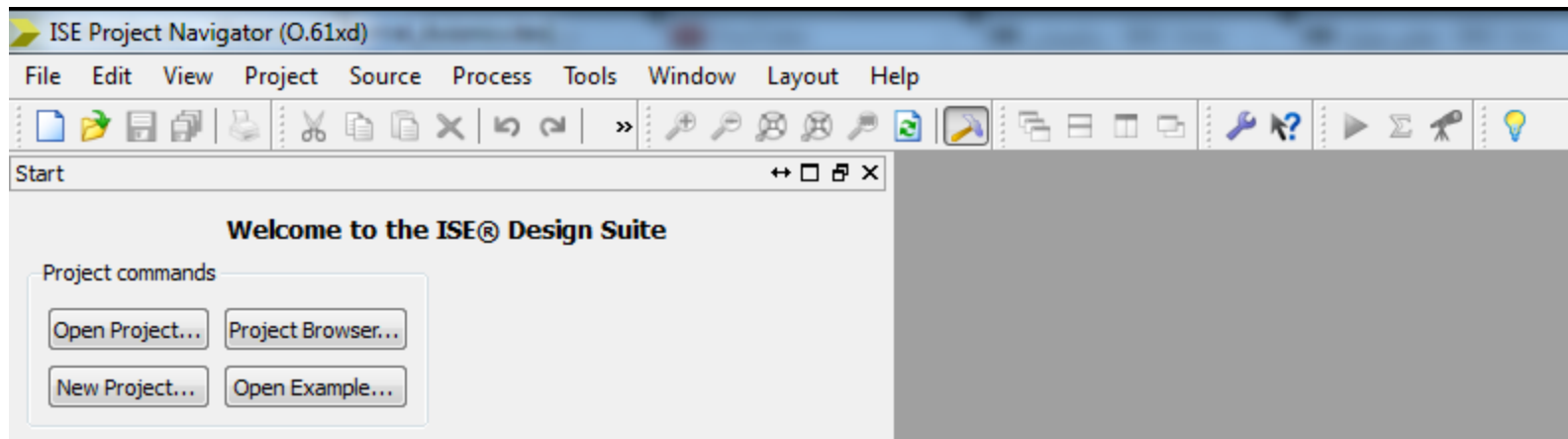
FPGAs Kit and Development Enviroment



How to Start the Hardware Development in FPGA



1. Open the Xilinx ISE project Navigator



2. Create a New Project and save it to appropriate location and please note that don't use spaces in the destination of the project

New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name:

Location: ...

Working Directory: ...

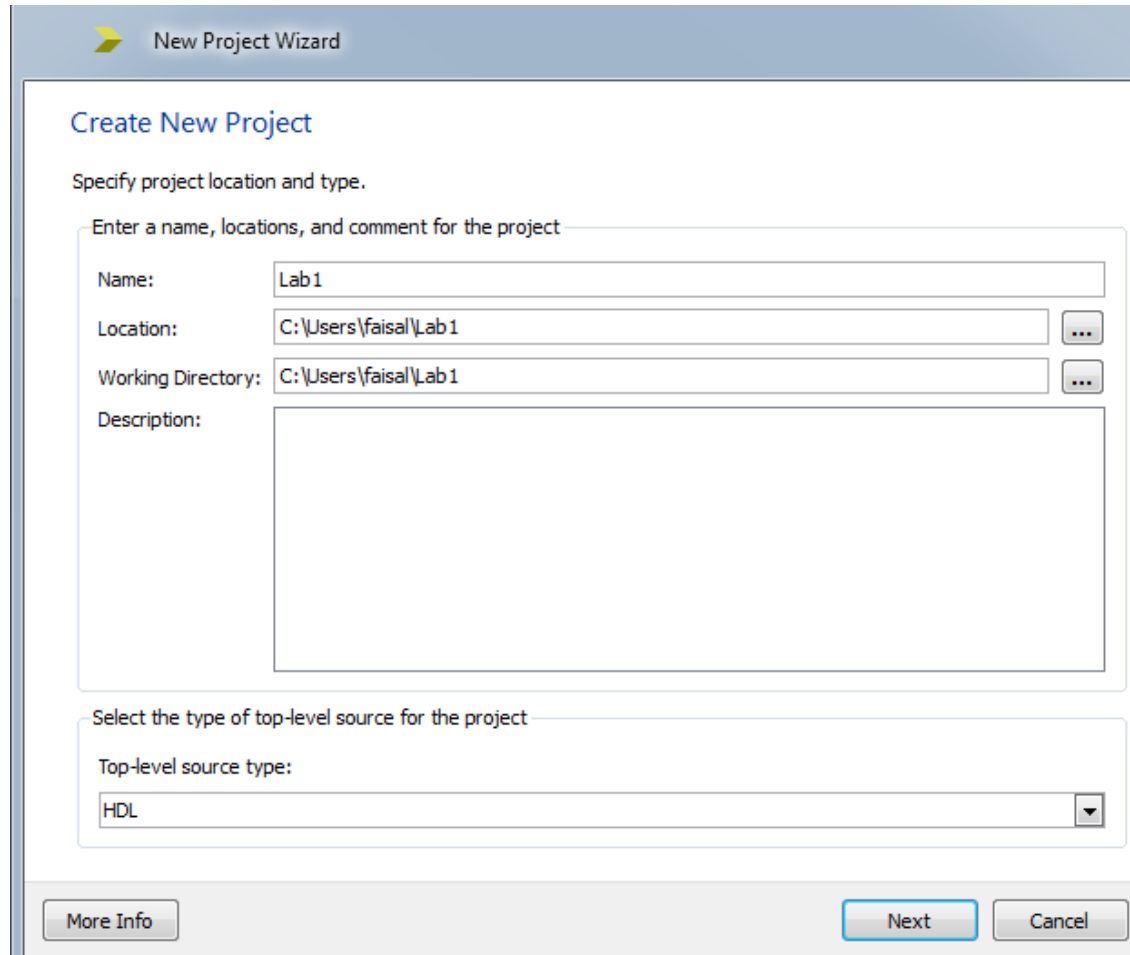
Description:

Select the type of top-level source for the project

Top-level source type:



3. Give the name of the project and click Next.



The image shows a 'New Project Wizard' dialog box. The title bar says 'New Project Wizard' with a yellow arrow icon. The main heading is 'Create New Project'. Below it, the instruction 'Specify project location and type.' is displayed. The first section, 'Enter a name, locations, and comment for the project', contains four fields: 'Name' (text box with 'Lab1'), 'Location' (text box with 'C:\Users\faisal\Lab1' and a browse button '...'), 'Working Directory' (text box with 'C:\Users\faisal\Lab1' and a browse button '...'), and 'Description' (a large empty text area). The second section, 'Select the type of top-level source for the project', contains a 'Top-level source type:' label and a dropdown menu currently showing 'HDL'. At the bottom, there are three buttons: 'More Info', 'Next' (highlighted with a blue border), and 'Cancel'.

New Project Wizard

Create New Project

Specify project location and type.

Enter a name, locations, and comment for the project

Name:

Location: ...

Working Directory: ...

Description:

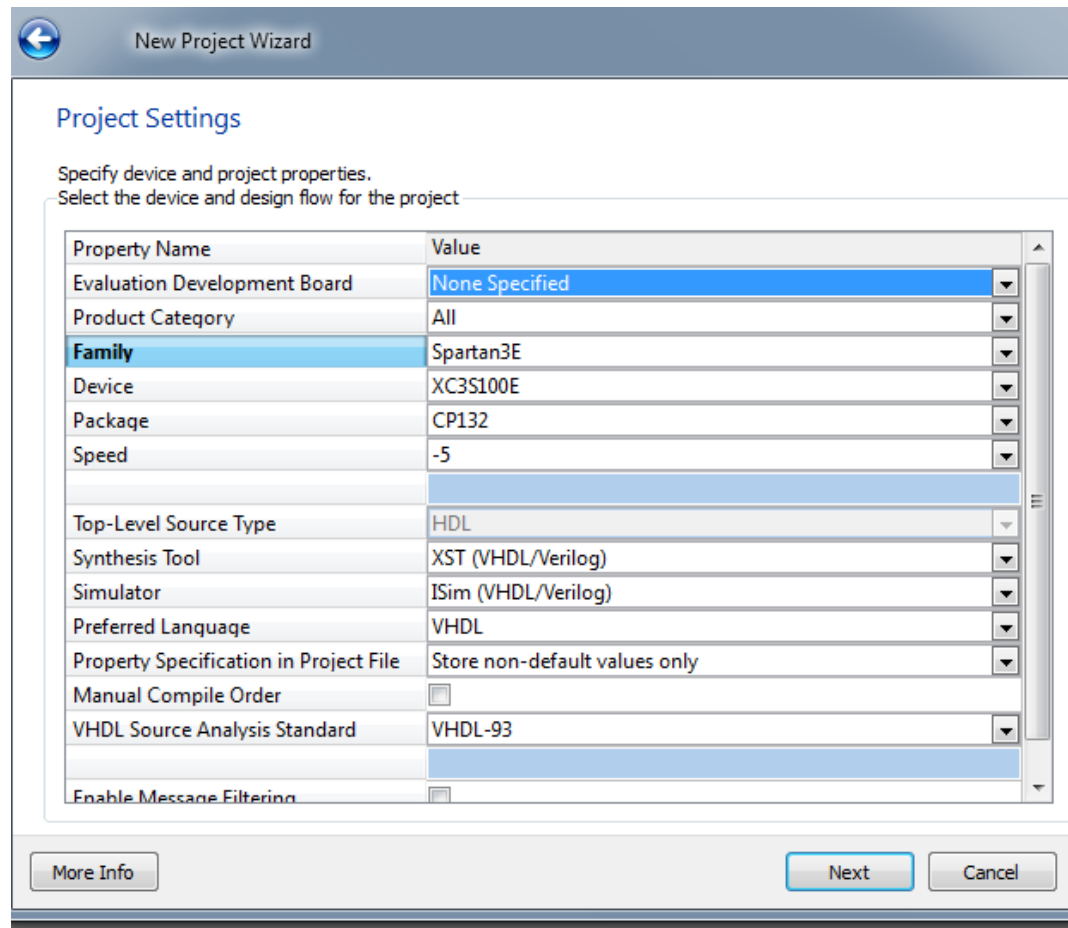
Select the type of top-level source for the project

Top-level source type:

More Info Next Cancel



4. Select the Family, Device, Package, Simulator and Preferred Language as shown below.



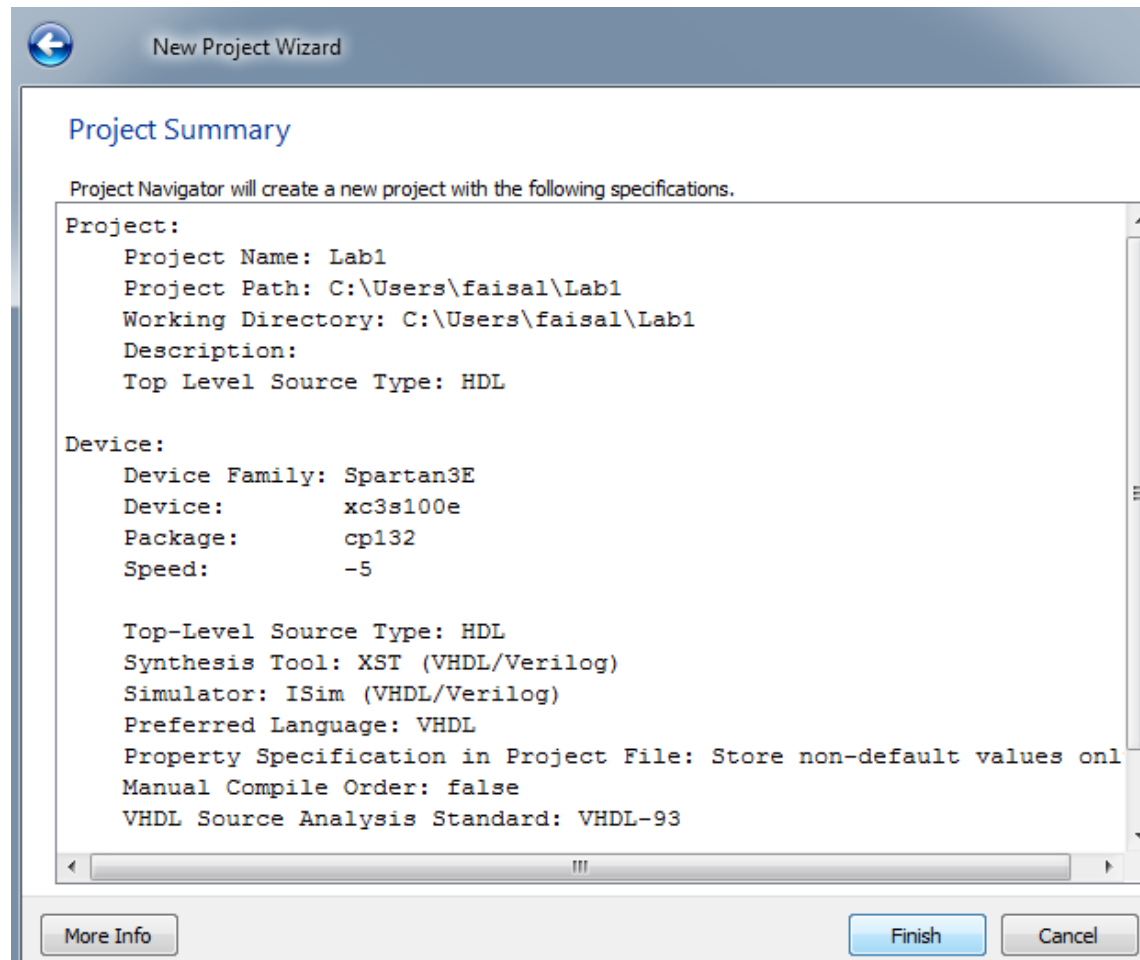
The image shows a 'New Project Wizard' dialog box with a 'Project Settings' tab. The settings are as follows:

Property Name	Value
Evaluation Development Board	None Specified
Product Category	All
Family	Spartan3E
Device	XC3S100E
Package	CP132
Speed	-5
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store non-default values only
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>

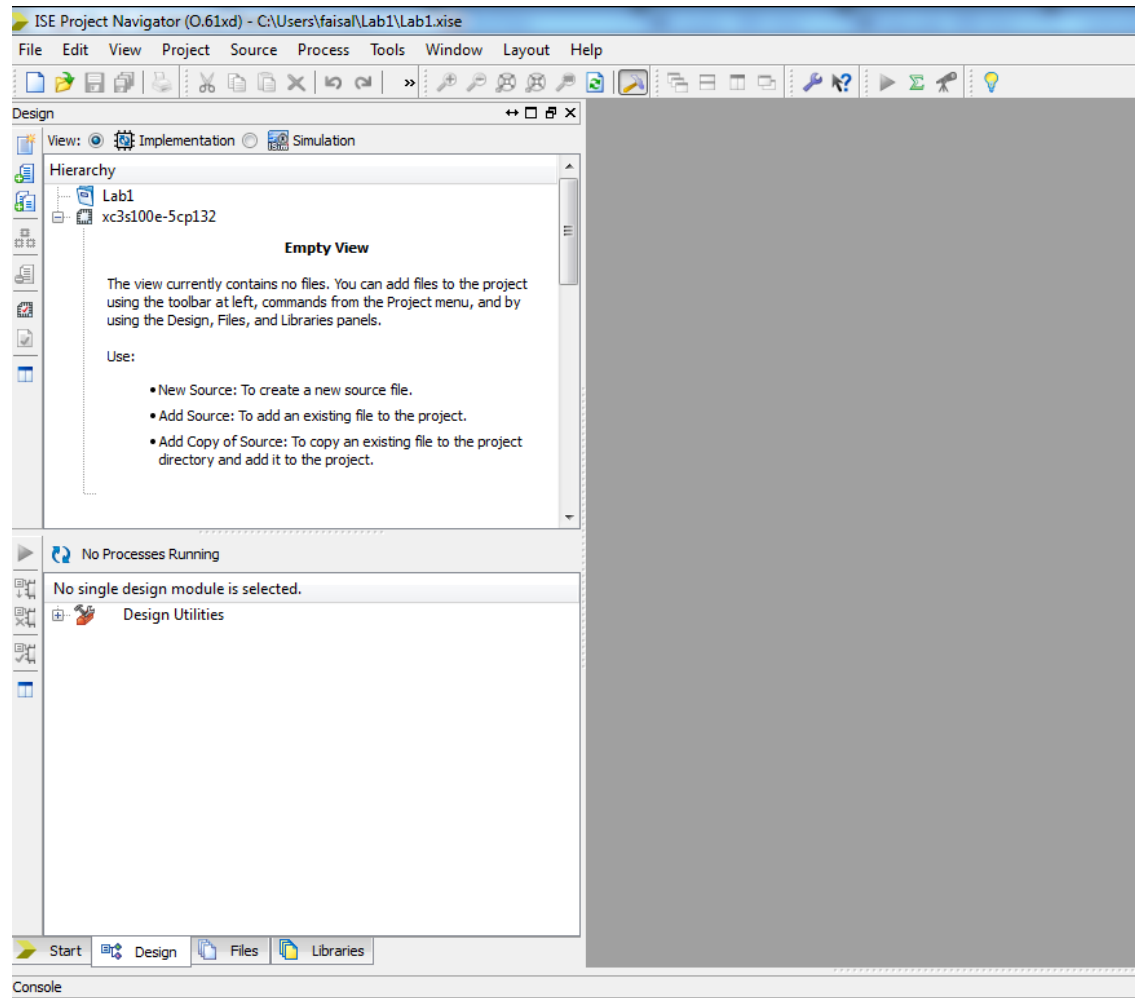
At the bottom of the dialog, there are three buttons: 'More Info', 'Next', and 'Cancel'.



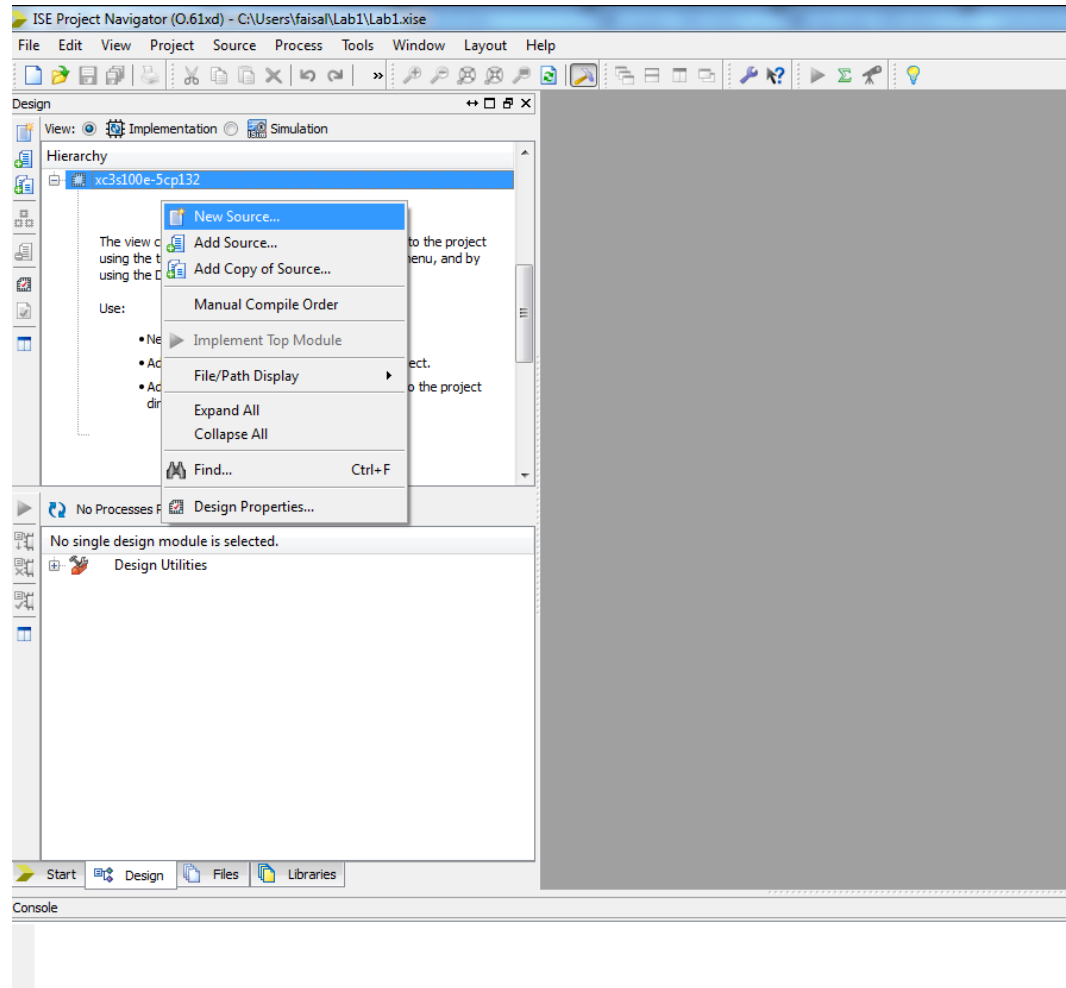
5. Click Finish.



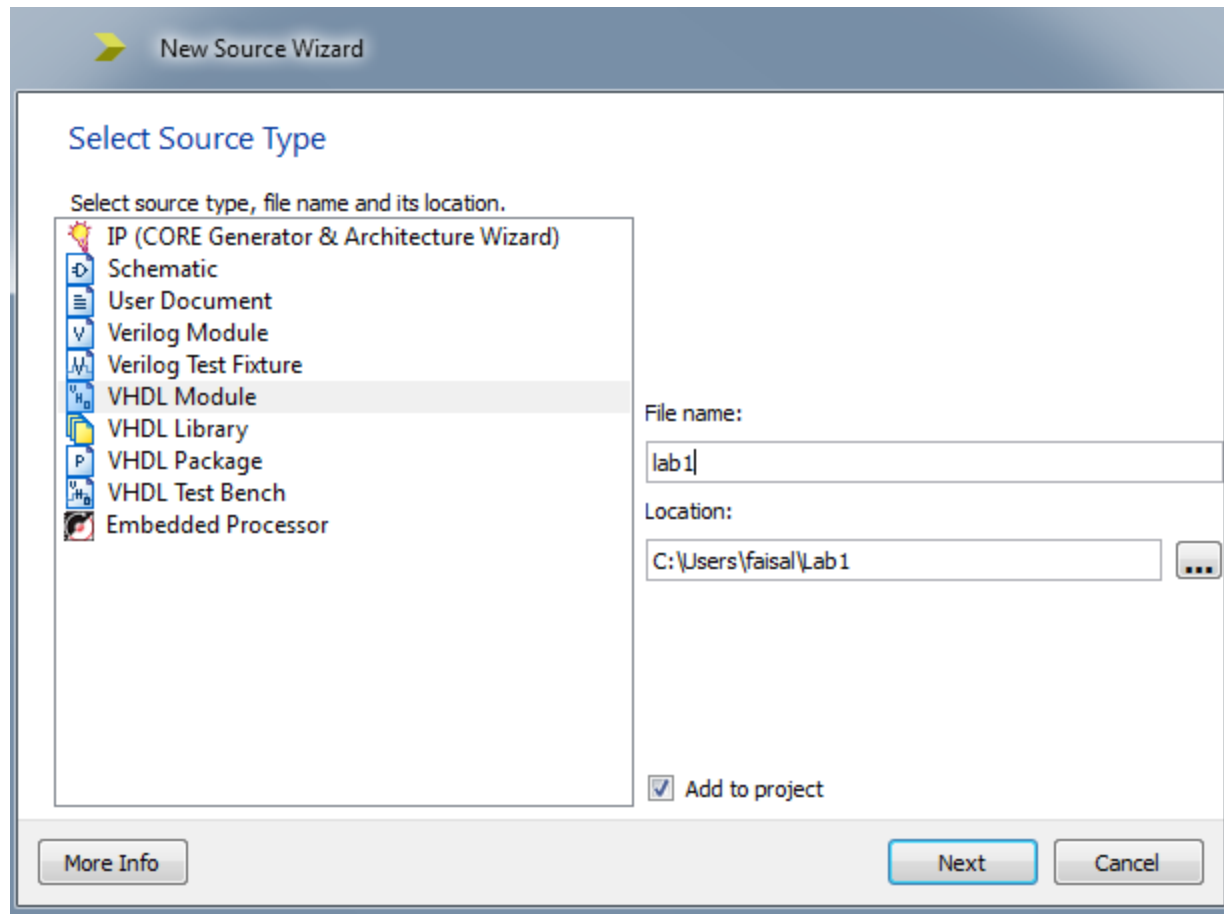
6. Now you have to Add a new vhdl file to write the code



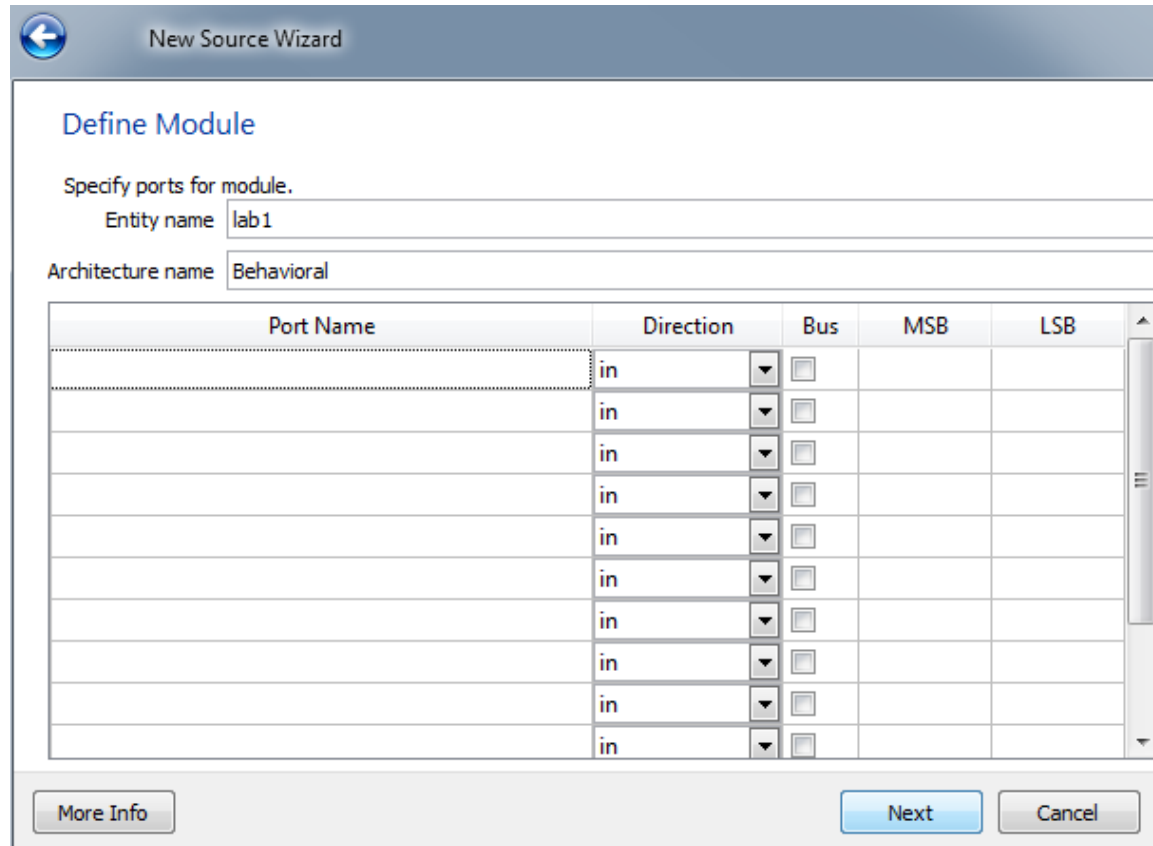
7. Right Click the on the name of the device.



4. Select the VHDL module as Source type and name the VHDL file



8. Click Next



The image shows a 'New Source Wizard' dialog box with a 'Define Module' section. It includes input fields for 'Entity name' (lab1) and 'Architecture name' (Behavioral). Below these is a table for defining ports. The table has columns for Port Name, Direction, Bus, MSB, and LSB. The first row is selected, and the Direction is set to 'in'. There are 10 rows in total. At the bottom, there are 'More Info', 'Next', and 'Cancel' buttons.

Define Module

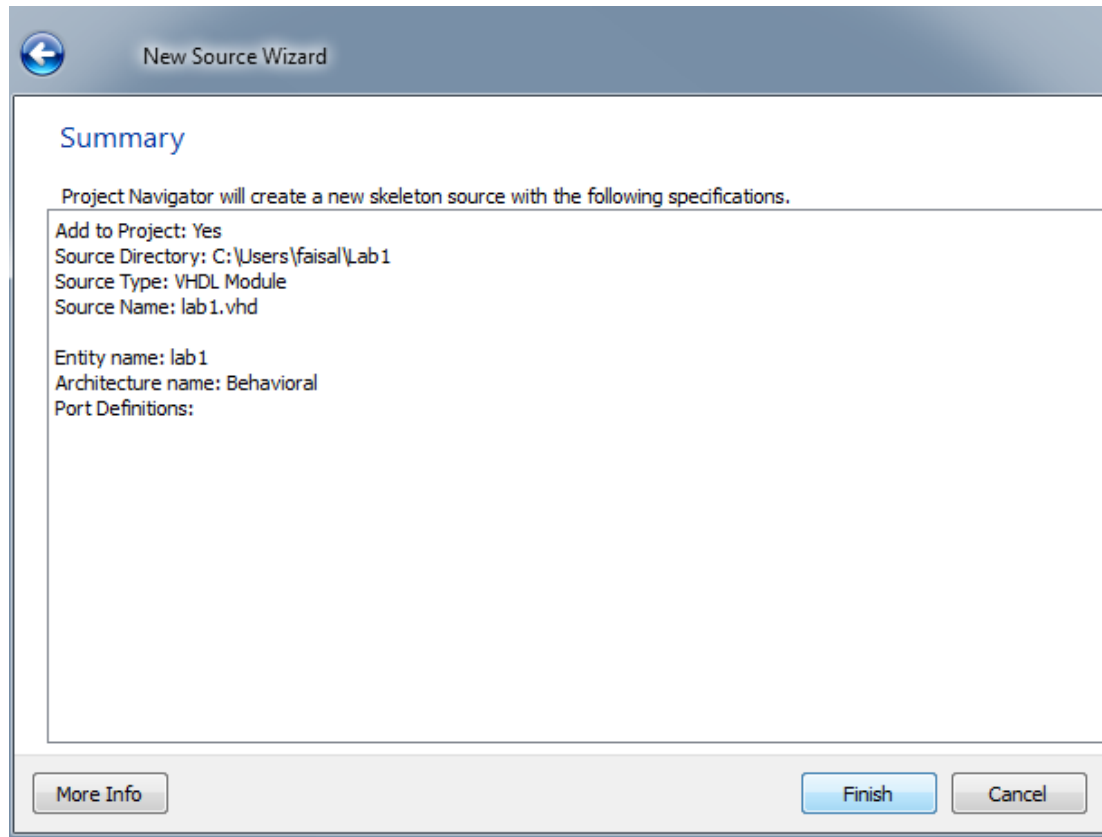
Specify ports for module.

Entity name

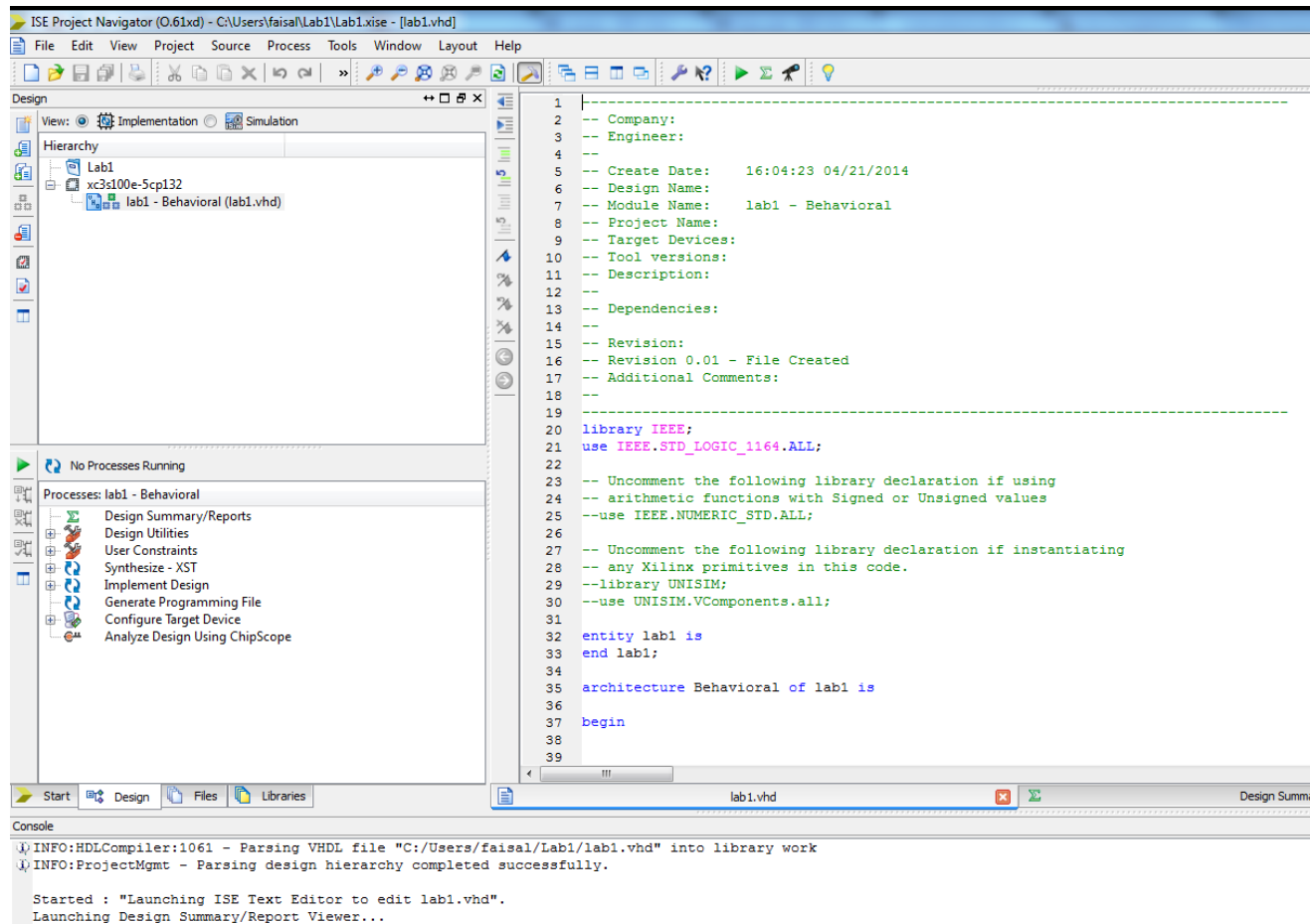
Architecture name

Port Name	Direction	Bus	MSB	LSB
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		
	in	<input type="checkbox"/>		

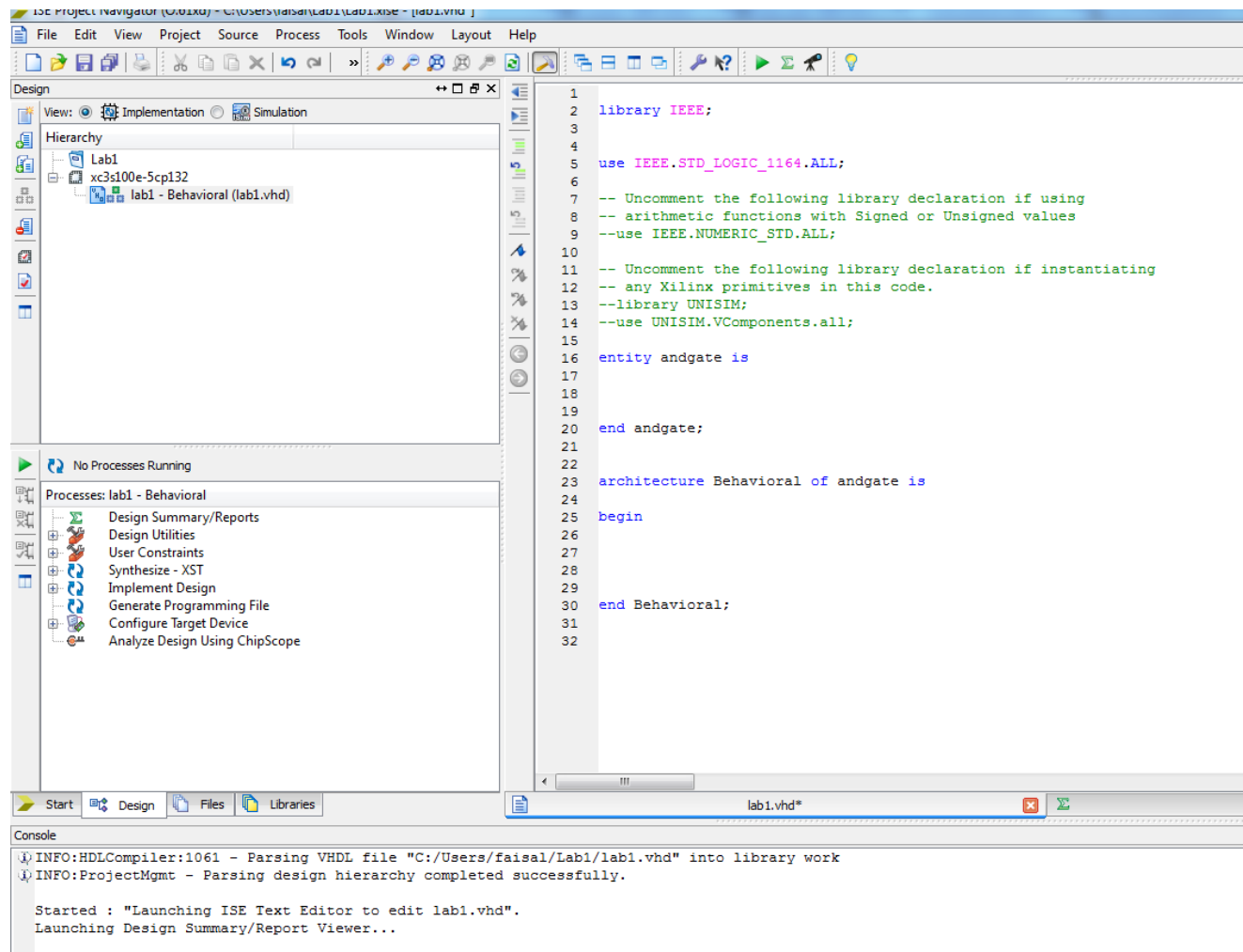




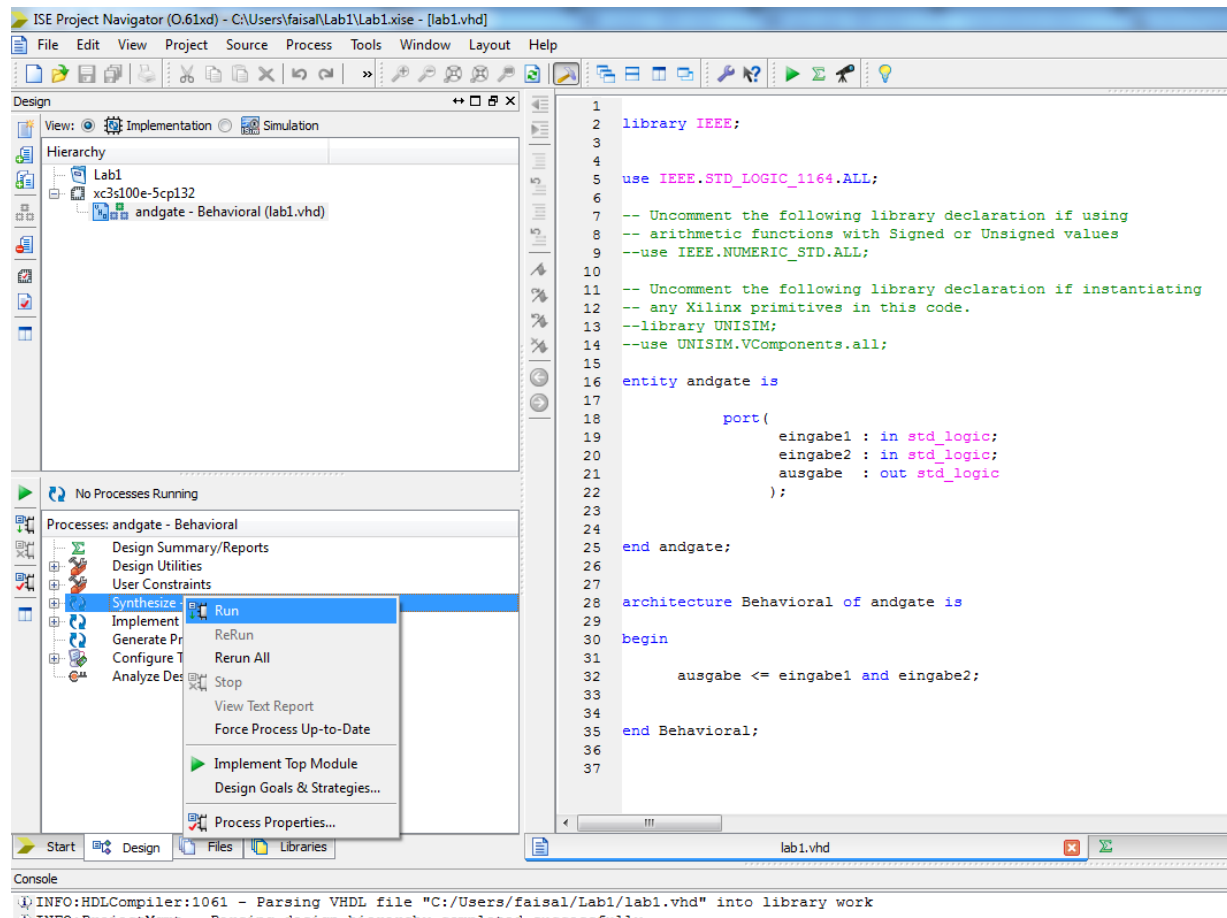
9. Open the VHDL file in the editor.



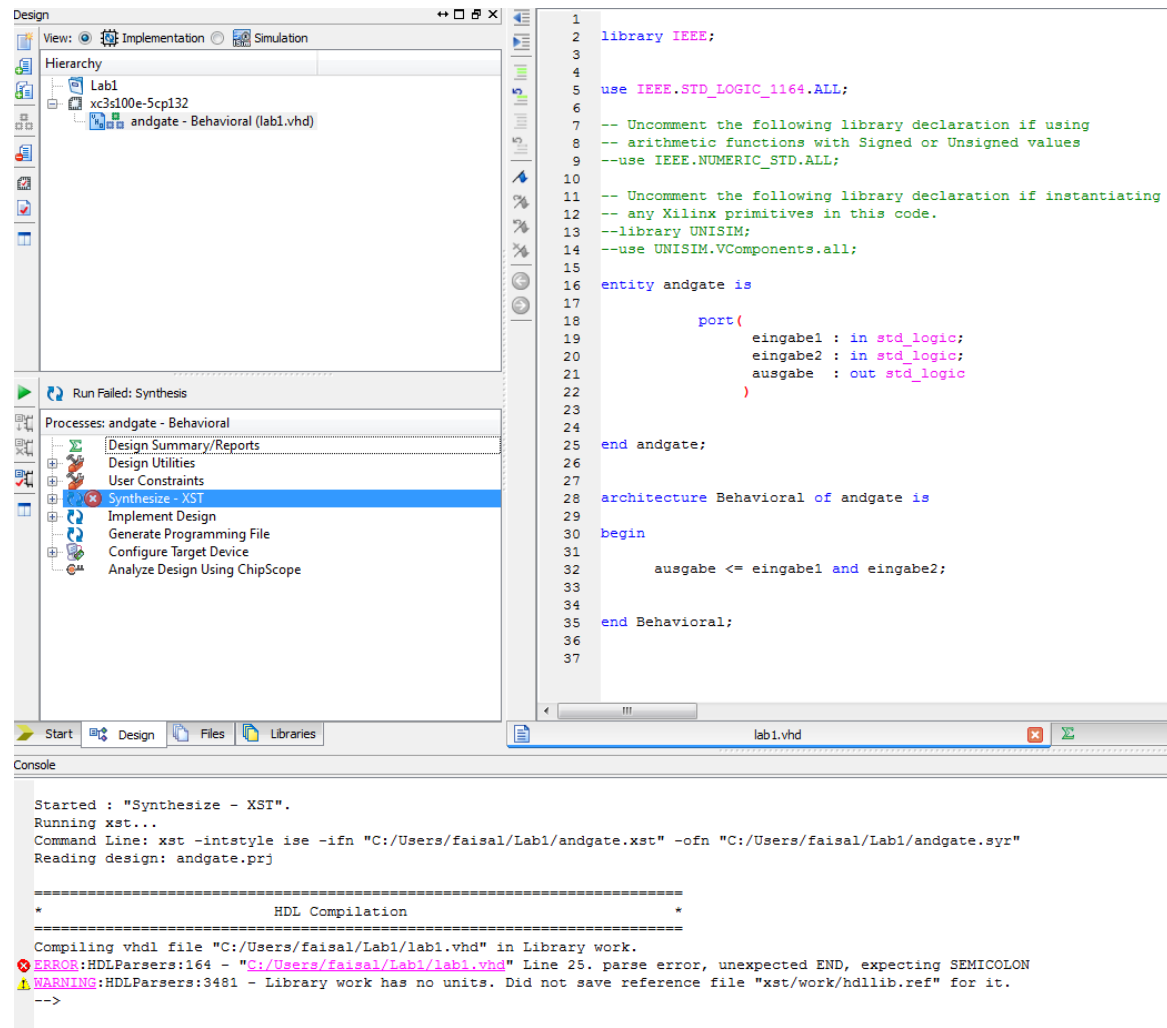
10. You can write your VHDL code here .



11. Write click on Synthesis option and select run.



12.If you have some error in your VHDL code it will be Red otherwise green.



13. Synthesis Process is Finished.

The screenshot displays the Xilinx ISE software interface. The top-left pane shows the project hierarchy with 'xc3s100e-5cp132' and 'andgate - Behavioral (lab1.vhd)'. The top-right pane shows the VHDL code for 'andgate.vhd'.

```
6
7  -- Uncomment the following library declaration if using
8  -- arithmetic functions with Signed or Unsigned values
9  --use IEEE.NUMERIC_STD.ALL;
10
11 -- Uncomment the following library declaration if instantiating
12 -- any Xilinx primitives in this code.
13 --library UNISIM;
14 --use UNISIM.VComponents.all;
15
16 entity andgate is
17
18     port(
19         eingabe1 : in std_logic;
20         eingabe2 : in std_logic;
21         ausgabe  : out std_logic
22     );
23
24 end andgate;
25
26
27 architecture Behavioral of andgate is
28
29 begin
30
31     ausgabe <= eingabe1 and eingabe2;
32
33
34 end Behavioral;
35
36
37
```

The bottom-left pane shows the 'Processes' list, with 'Synthesize - XST' highlighted. The bottom-right pane shows the 'Console' output, which includes the 'Final Report'.

Final Report

Clock Information:
No clock signals found in this design

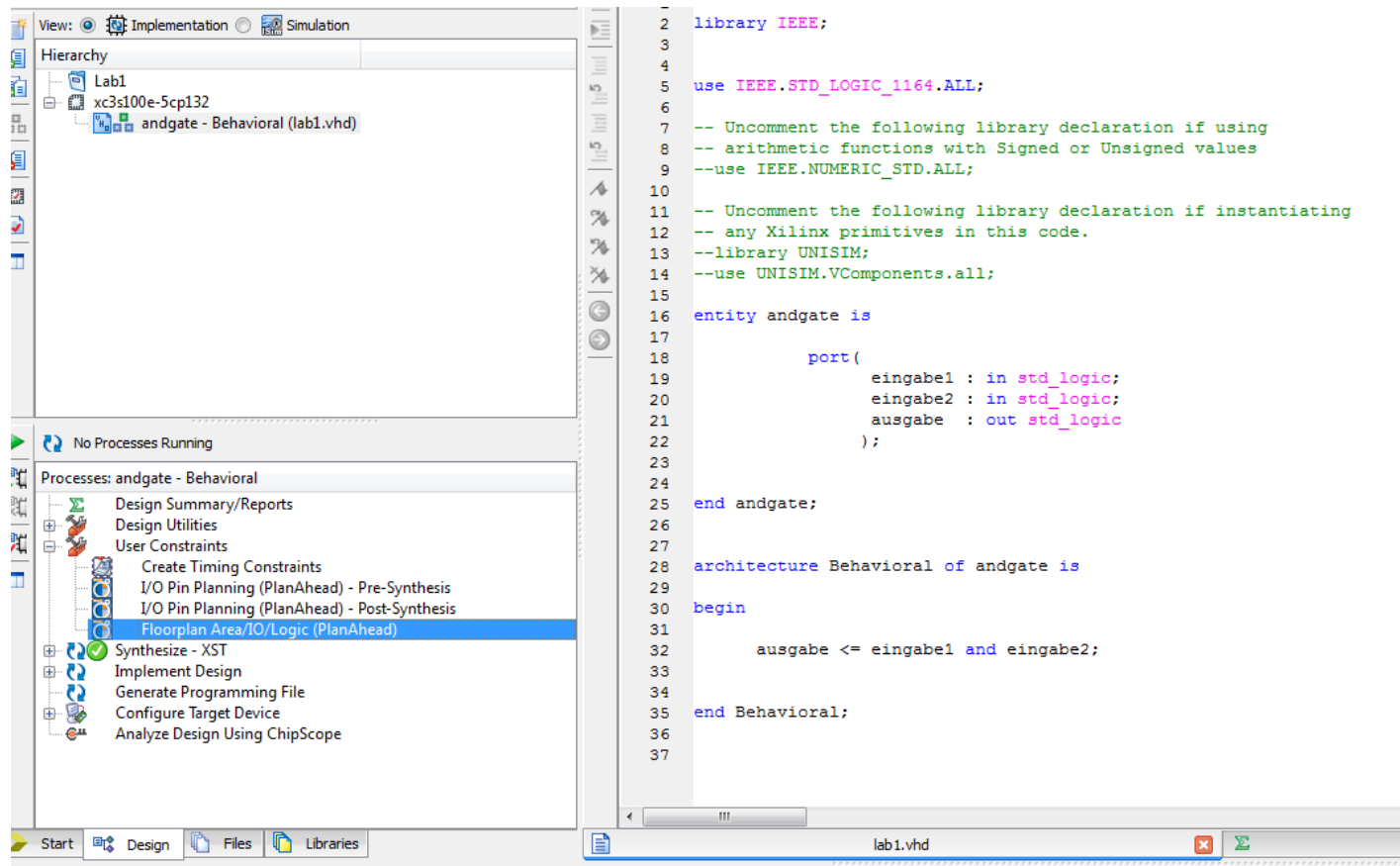
Asynchronous Control Signals Information:
No asynchronous control signals found in this design

Timing Summary:
Speed Grade: -5

Minimum period: No path found
Minimum input arrival time before clock: No path found
Maximum output required time after clock: No path found



14. Now Pin Assignment has to be done so Click on FloorPlanning

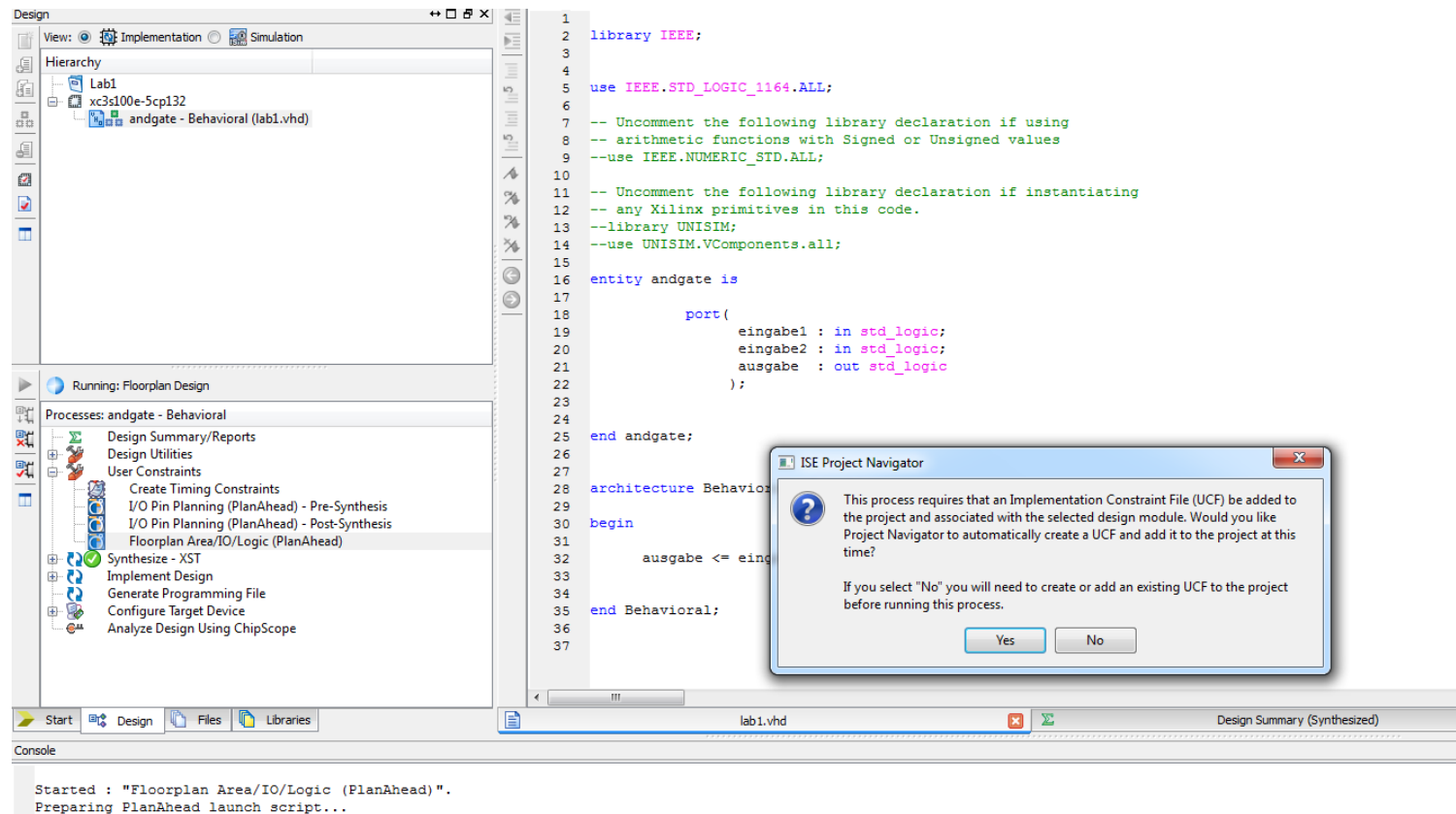


The screenshot displays the Xilinx ISE software interface. On the left, the 'Processes' window for 'andgate - Behavioral' is open, showing a list of design steps. The 'Floorplan Area/IO/Logic (PlanAhead)' step is highlighted with a blue selection bar. The main window on the right shows the VHDL code for 'lab1.vhd'. The code defines an 'andgate' entity with two input ports ('eingabe1' and 'eingabe2') and one output port ('ausgabe'). The behavior is implemented as a simple AND gate logic: 'ausgabe <= eingabe1 and eingabe2;'.

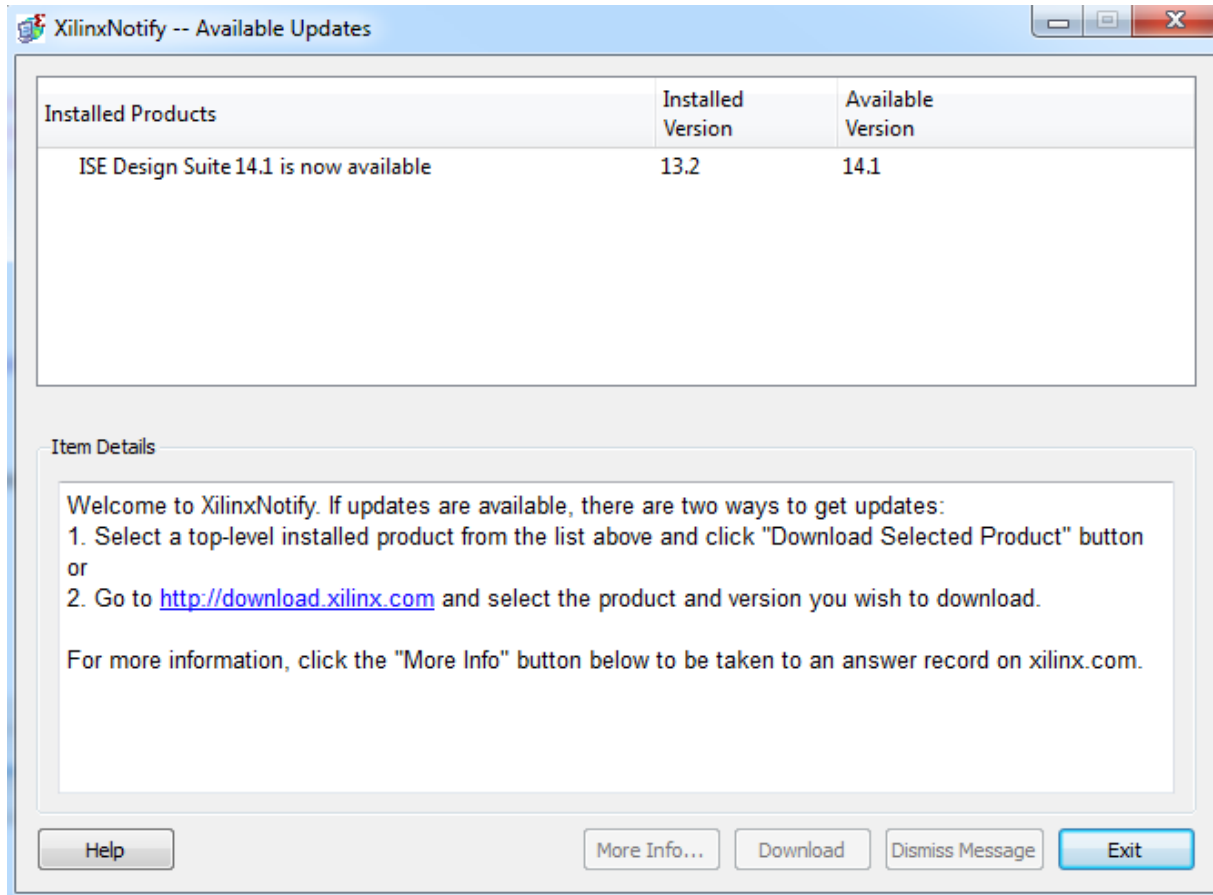
```
1 library IEEE;
2
3
4 use IEEE.STD_LOGIC_1164.ALL;
5
6
7 -- Uncomment the following library declaration if using
8 -- arithmetic functions with Signed or Unsigned values
9 --use IEEE.NUMERIC_STD.ALL;
10
11 -- Uncomment the following library declaration if instantiating
12 -- any Xilinx primitives in this code.
13 --library UNISIM;
14 --use UNISIM.VComponents.all;
15
16 entity andgate is
17
18     port(
19         eingabe1 : in std_logic;
20         eingabe2 : in std_logic;
21         ausgabe  : out std_logic
22     );
23
24 end andgate;
25
26
27 architecture Behavioral of andgate is
28
29 begin
30
31     ausgabe <= eingabe1 and eingabe2;
32
33 end Behavioral;
34
35
36
37
```



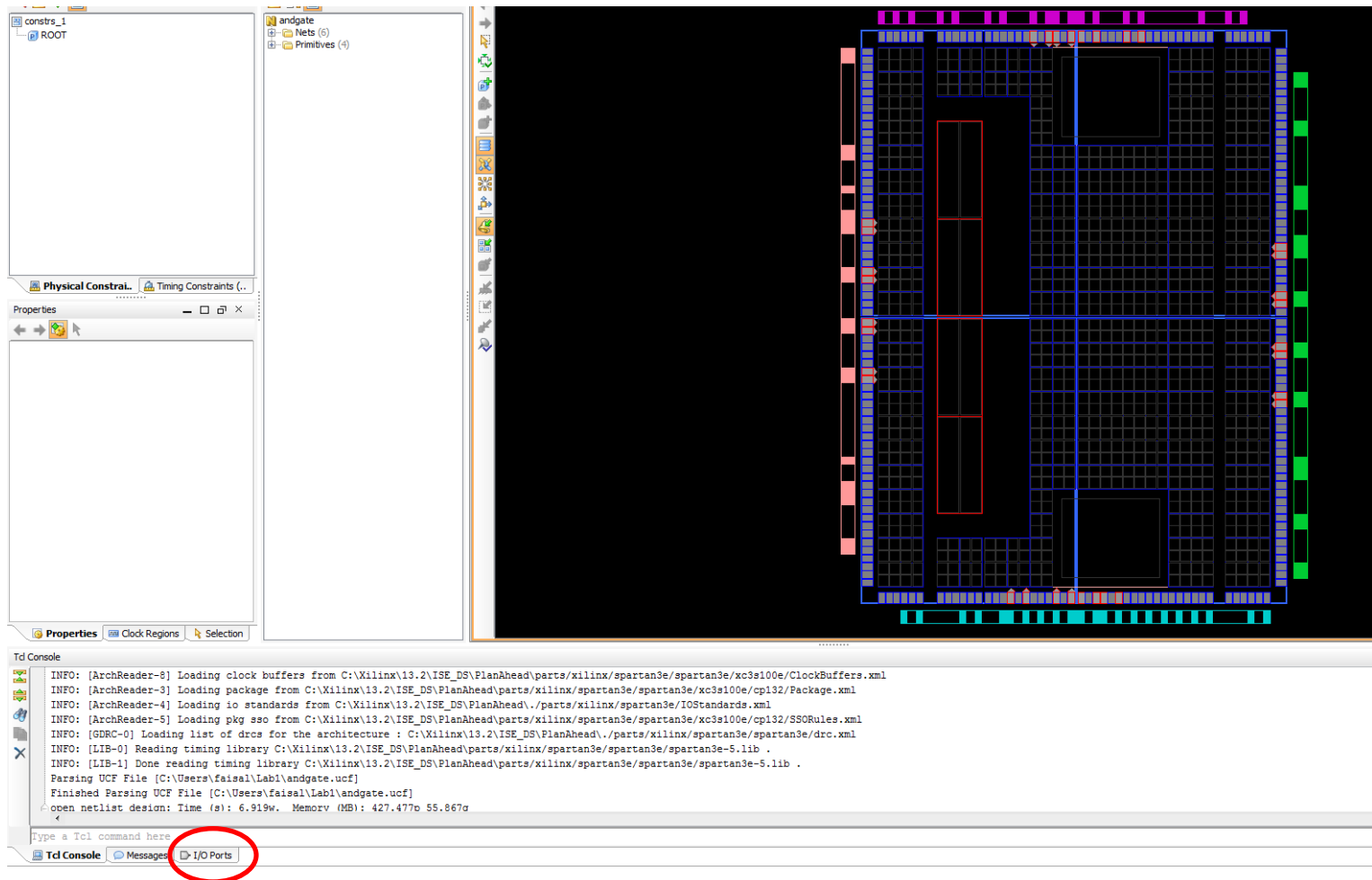
15. Click Yes and Wait , It takes some time to open the window for floor planning (Pin Assignment)



16. Don't update the software so press Exit button to cancel the update.



17. This is place where you will map your I/O's to the actual hardware



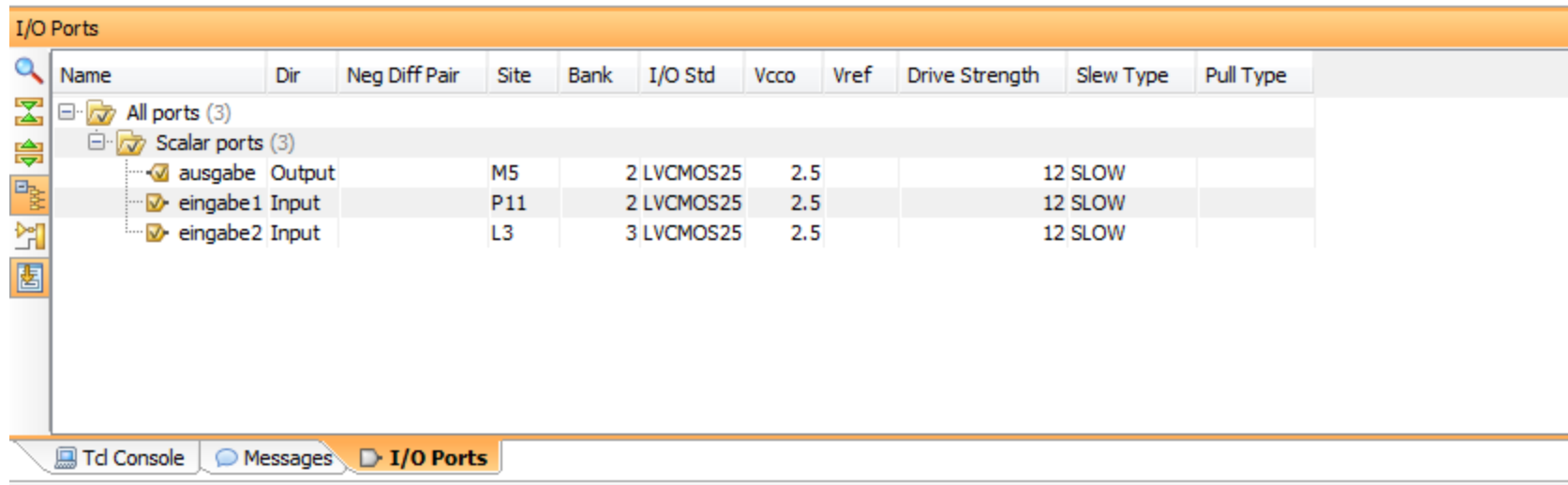
18. Click the I/O ports Tab

The screenshot displays the Netlist Design tool interface. The main window shows a grid-based layout of a circuit board with various components and nets. The left sidebar contains several panels: 'Physical Constraints', 'Netlist', 'Properties', and 'I/O Ports'. The 'I/O Ports' panel is currently selected and shows a table of port configurations.

Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (3)										
Scalar ports (3)										
ausgabe Output					LVCMOS25	2.5		12 SLOW		
eingabe1 Input					LVCMOS25	2.5		12 SLOW		
eingabe2 Input					LVCMOS25	2.5		12 SLOW		



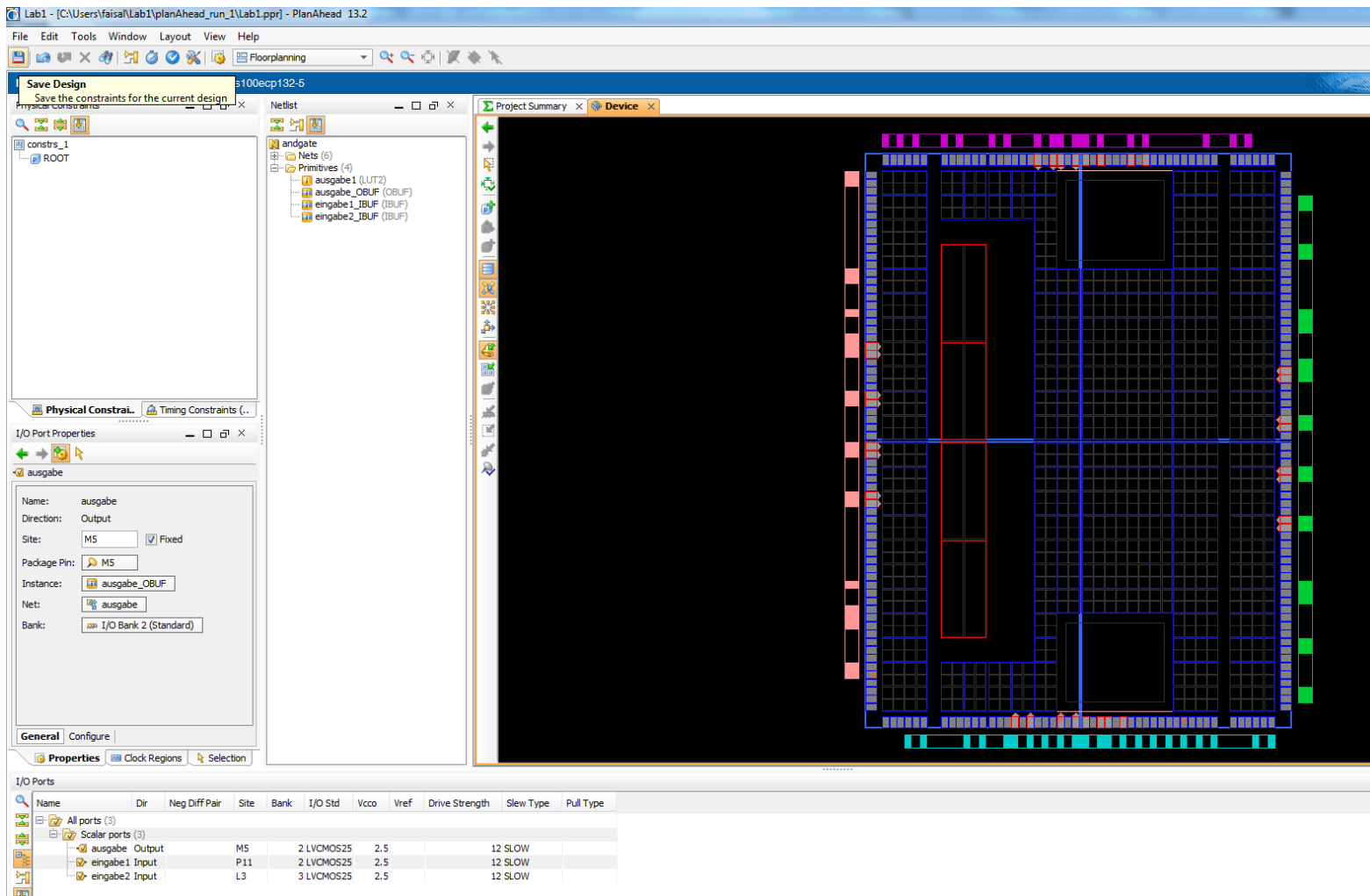
19. Select the Appropriate Pins for each of the Port Signal from the Site Column by double clicking it corresponding to the Signal.



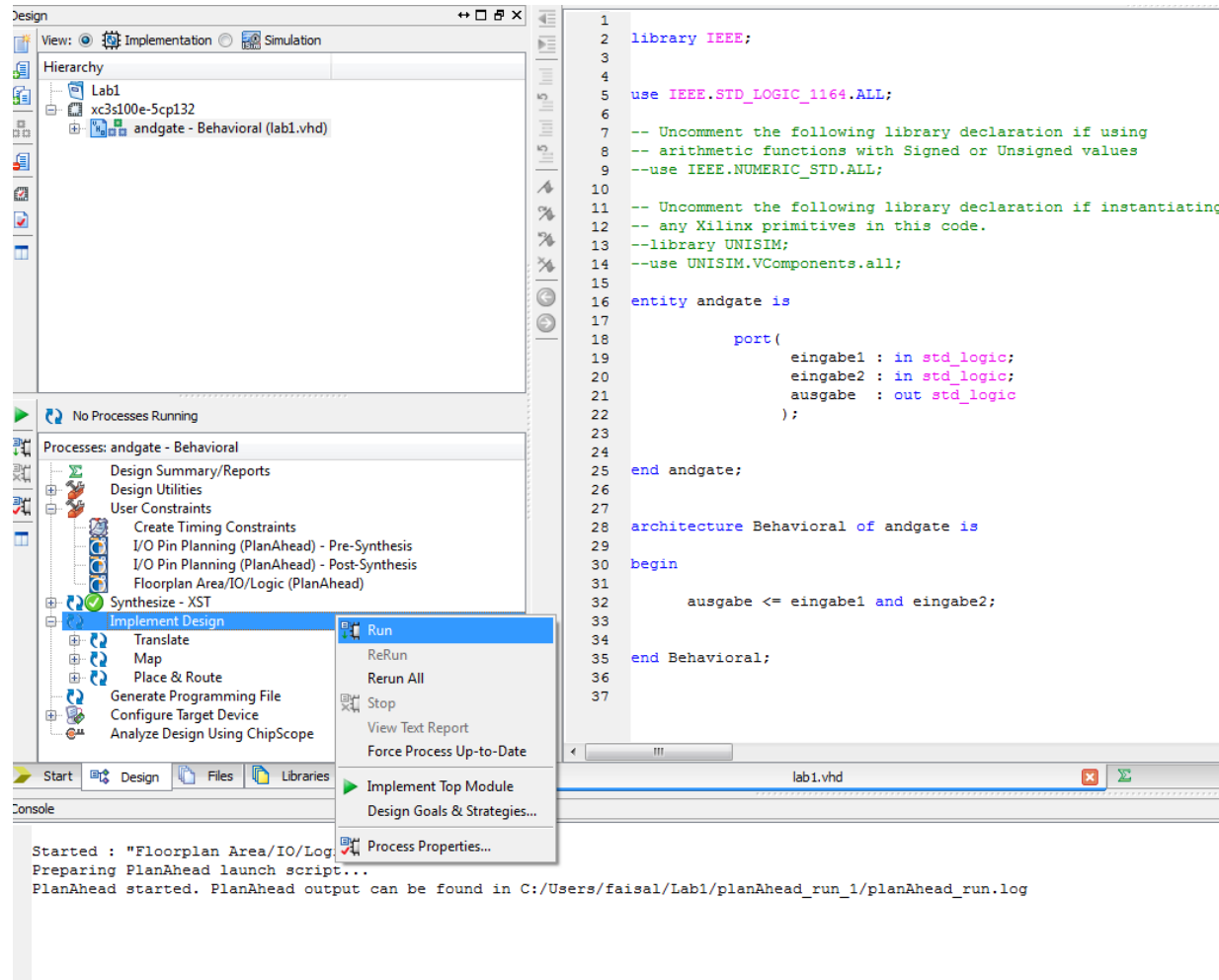
Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (3)										
Scalar ports (3)										
ausgabe	Output		M5	2	LVCMOS25	2.5			12 SLOW	
eingabe1	Input		P11	2	LVCMOS25	2.5			12 SLOW	
eingabe2	Input		L3	3	LVCMOS25	2.5			12 SLOW	



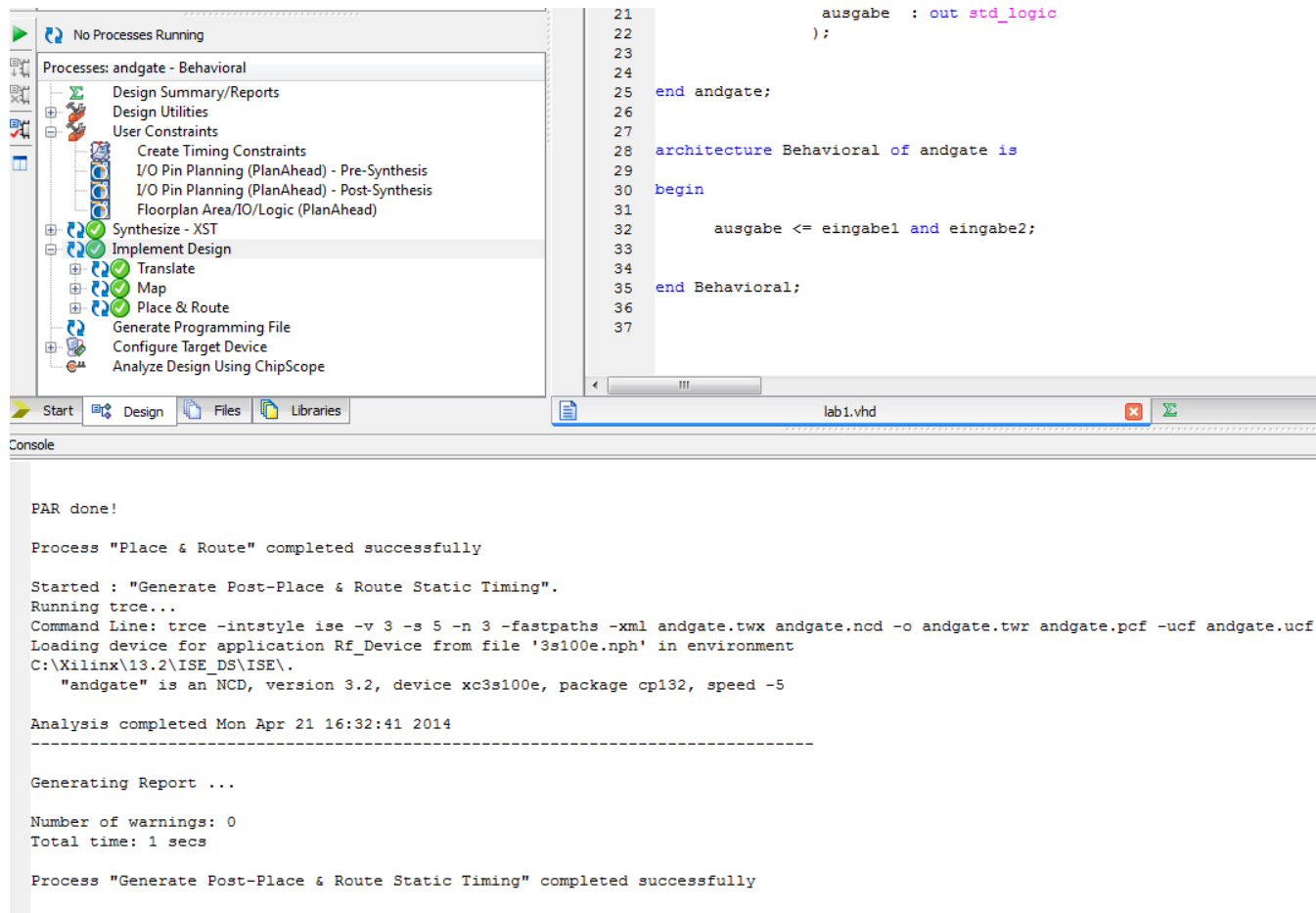
20a. When you are done with the pin Assignment , Click to save the changes and switch to the ISE design Software.



20b. Now right click on the Implement design and click run to Translate, Map , Place and Route



21. When the design is implemented successfully it will be shown up green as follows.



The screenshot displays the Xilinx ISE software interface. On the left, the 'Processes' pane shows a list of design steps, with 'Implement Design' and its sub-steps ('Translate', 'Map', 'Place & Route') marked with green checkmarks, indicating successful completion. The main editor window shows a VHDL code snippet for an 'andgate' entity. The code defines two inputs ('eingabe1', 'eingabe2') and one output ('ausgabe'), with a behavioral architecture that implements a logical AND function. The console window at the bottom provides a detailed log of the implementation process, including the command line used for synthesis and placement, the device configuration (xc3s100e), and the final status: 'Analysis completed Mon Apr 21 16:32:41 2014' and 'Generating Report ...'. The console also reports 'Number of warnings: 0' and 'Total time: 1 secs'.

```
21         ausgabe : out std_logic
22         );
23
24
25     end andgate;
26
27
28     architecture Behavioral of andgate is
29
30     begin
31
32         ausgabe <= eingabe1 and eingabe2;
33
34
35     end Behavioral;
36
37
```

PAR done!

Process "Place & Route" completed successfully

Started : "Generate Post-Place & Route Static Timing".

Running trce...

Command Line: trce -intstyle ise -v 3 -s 5 -n 3 -fastpaths -xml andgate.twx andgate.ncd -o andgate.twr andgate.pcf -ucf andgate.ucf

Loading device for application Rf_Device from file '3s100e.nph' in environment

C:\Xilinx\13.2\ISE_DS\ISE\.

"andgate" is an NCD, version 3.2, device xc3s100e, package cp132, speed -5

Analysis completed Mon Apr 21 16:32:41 2014

Generating Report ...

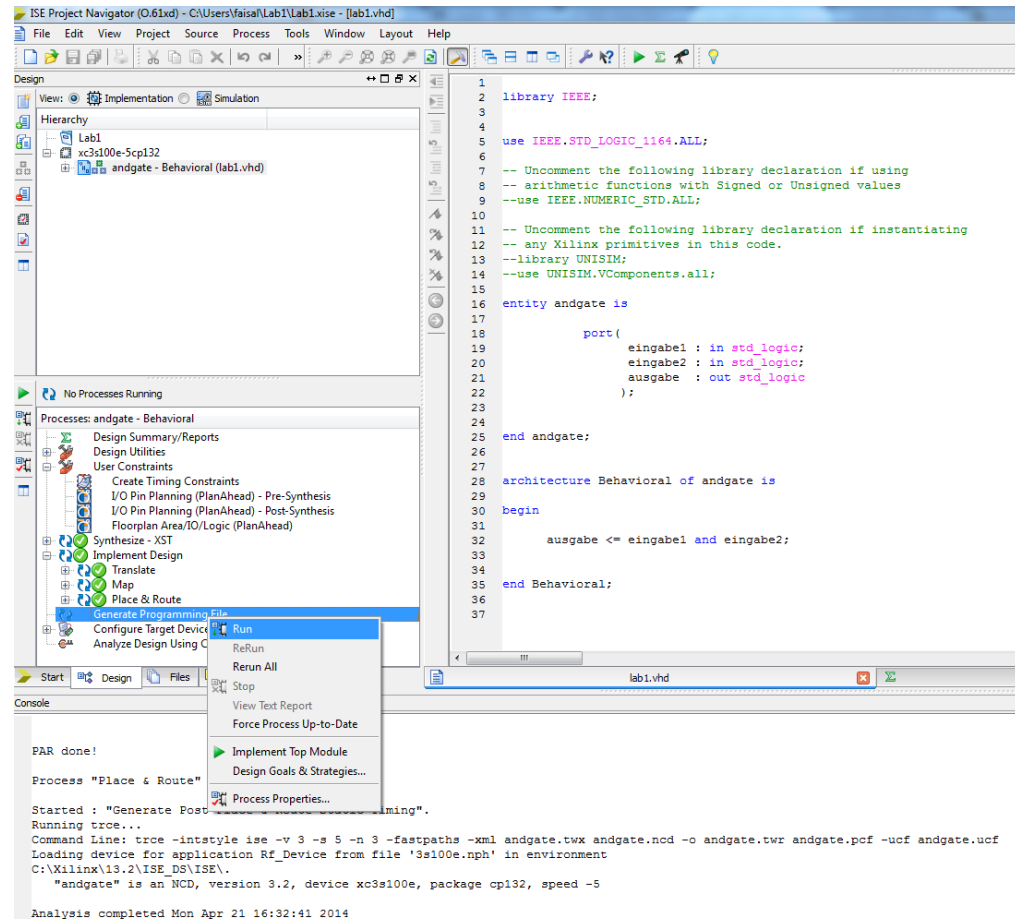
Number of warnings: 0

Total time: 1 secs

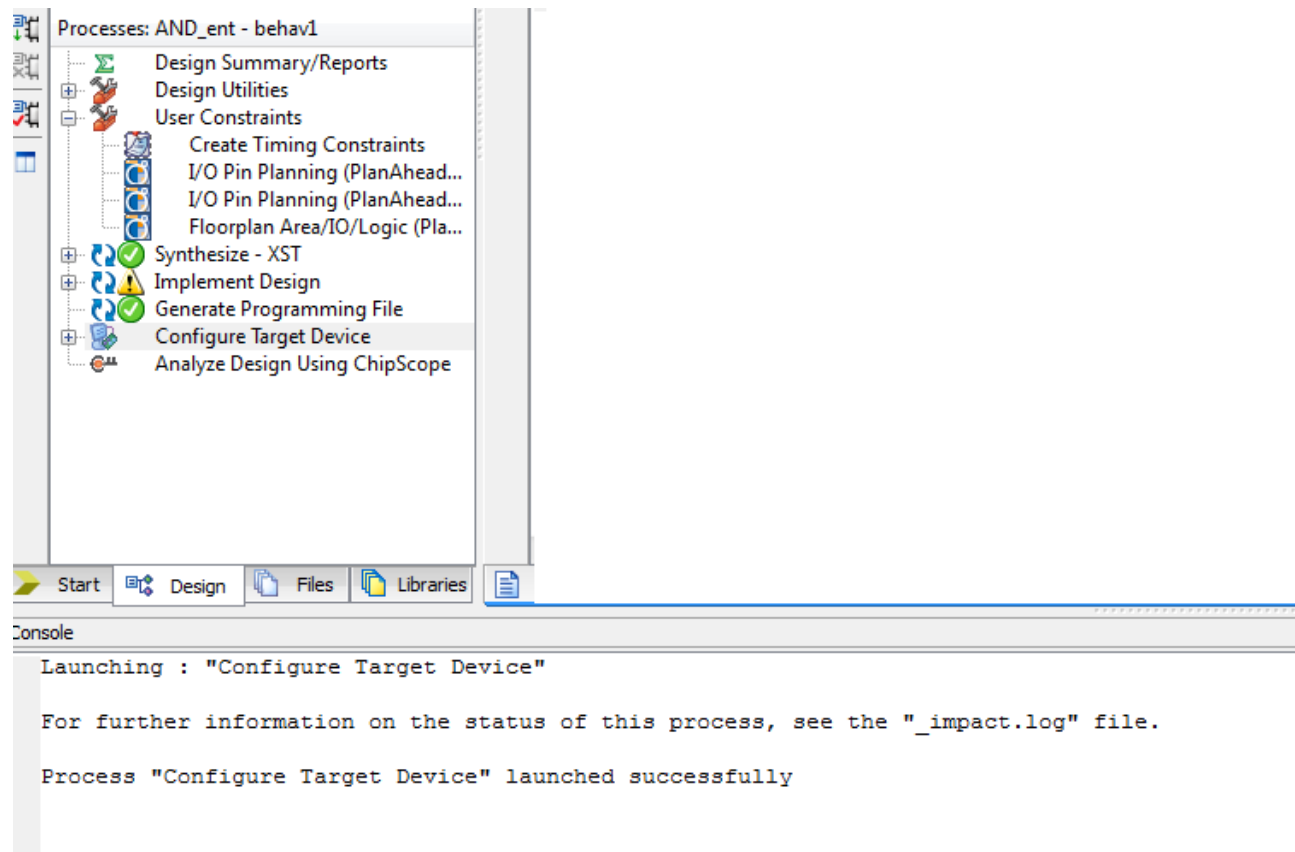
Process "Generate Post-Place & Route Static Timing" completed successfully



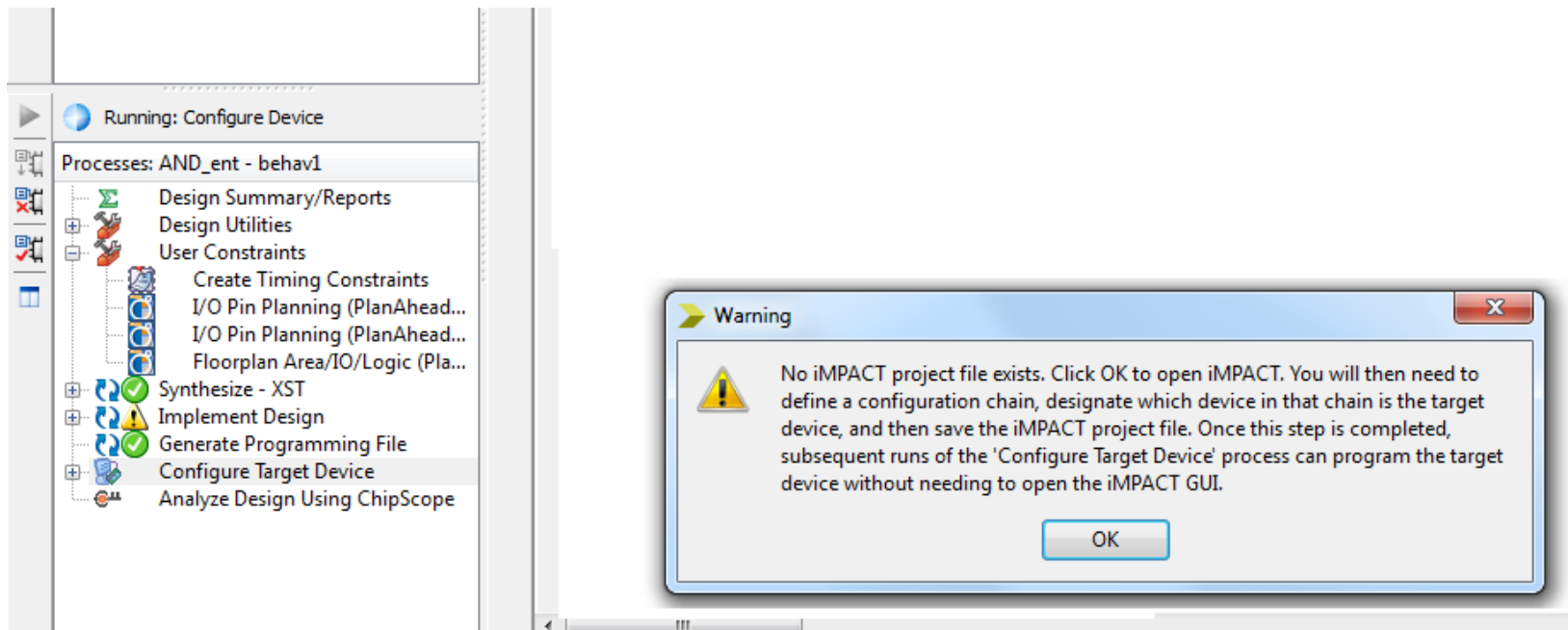
20. Now you have to Generate the programming File so right click on it and run.



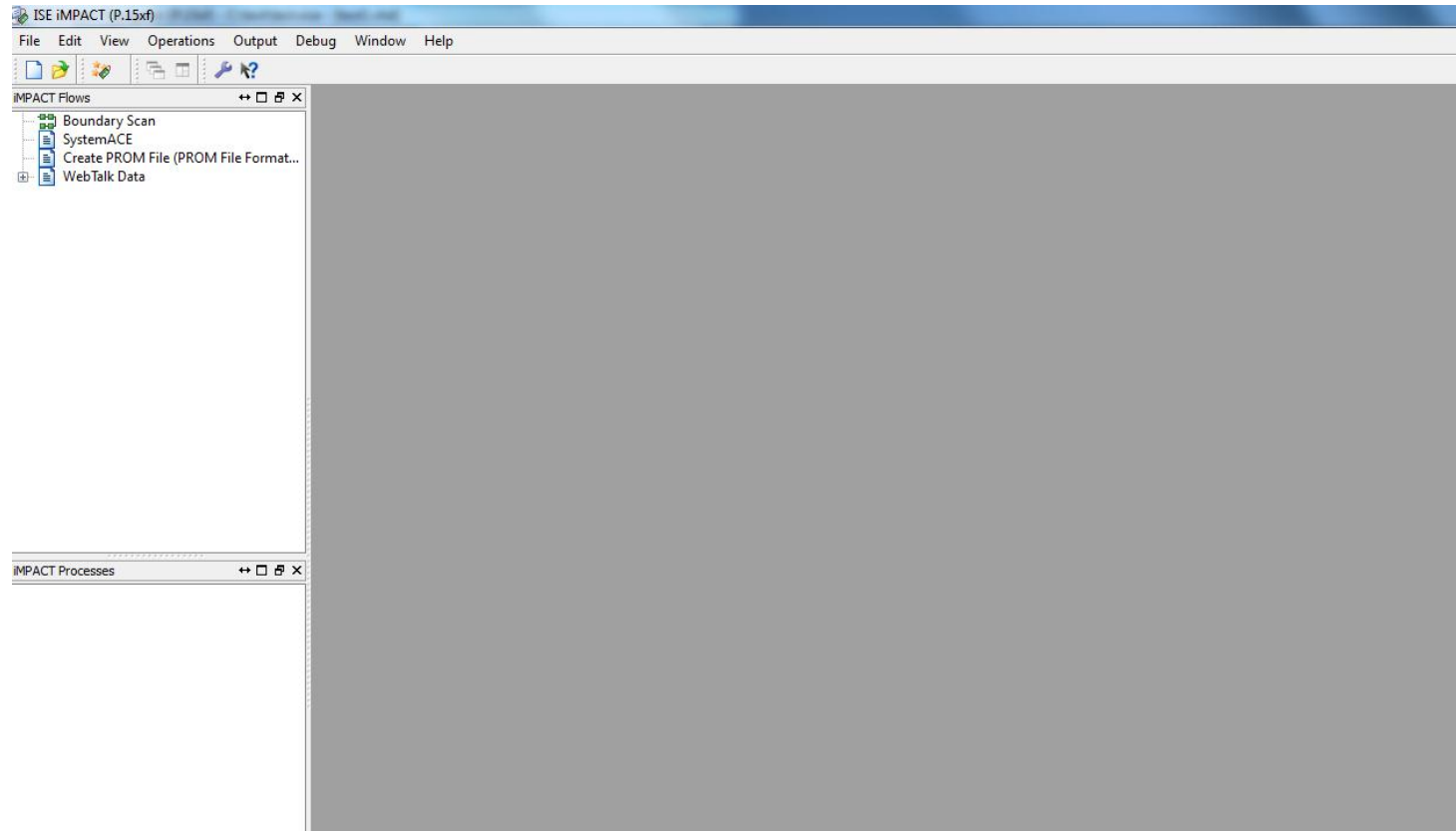
22. If this process is successfully completed then you will have a .bit file in your project directory which will be downloaded on the FPGA kit.



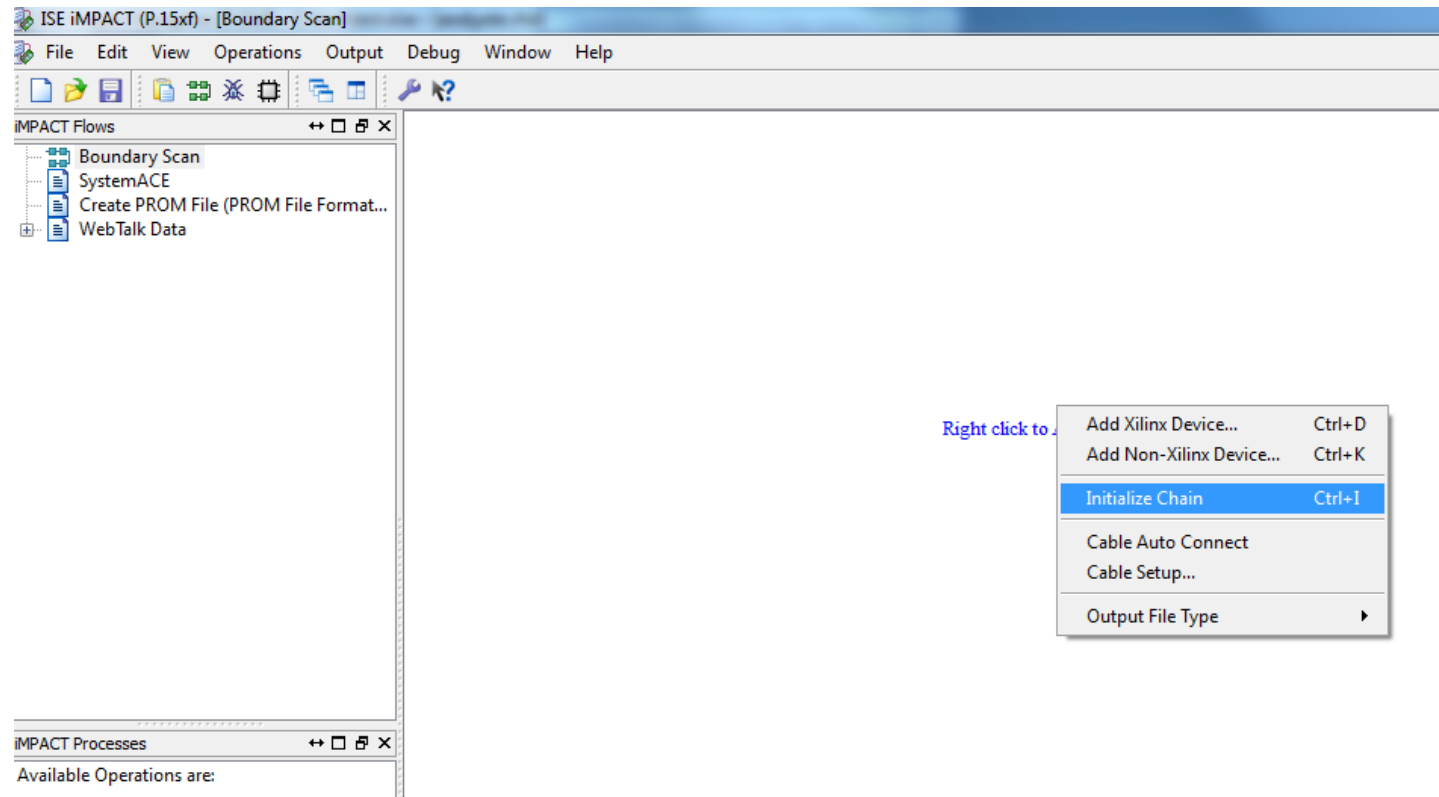
22. Now it's the turn of the downloading procedure so double click on Configure Target Device.



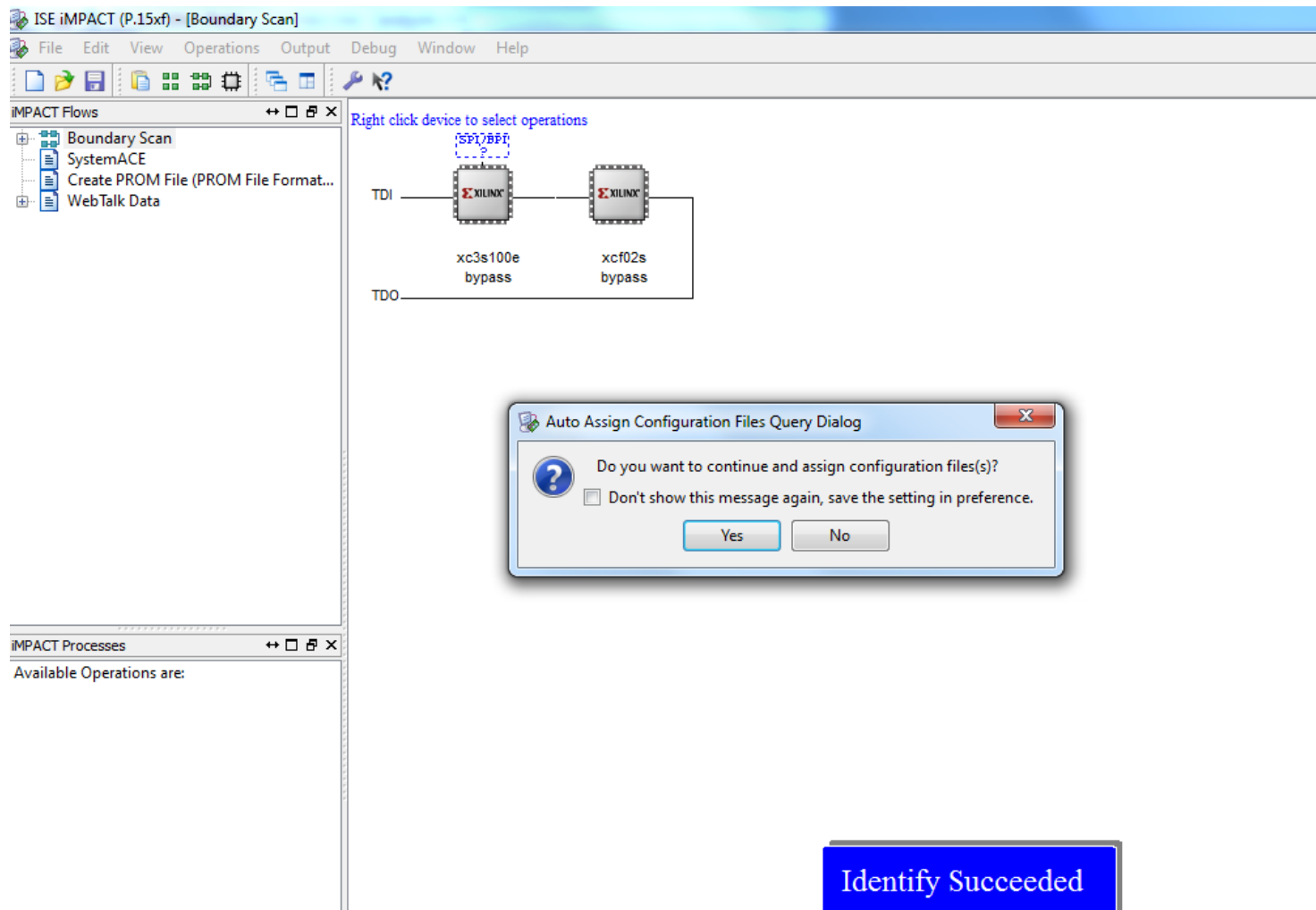
23. Double Click on Boundary Scan.



24a. Follow the instruction on the white window and select the Initialize Chain option.



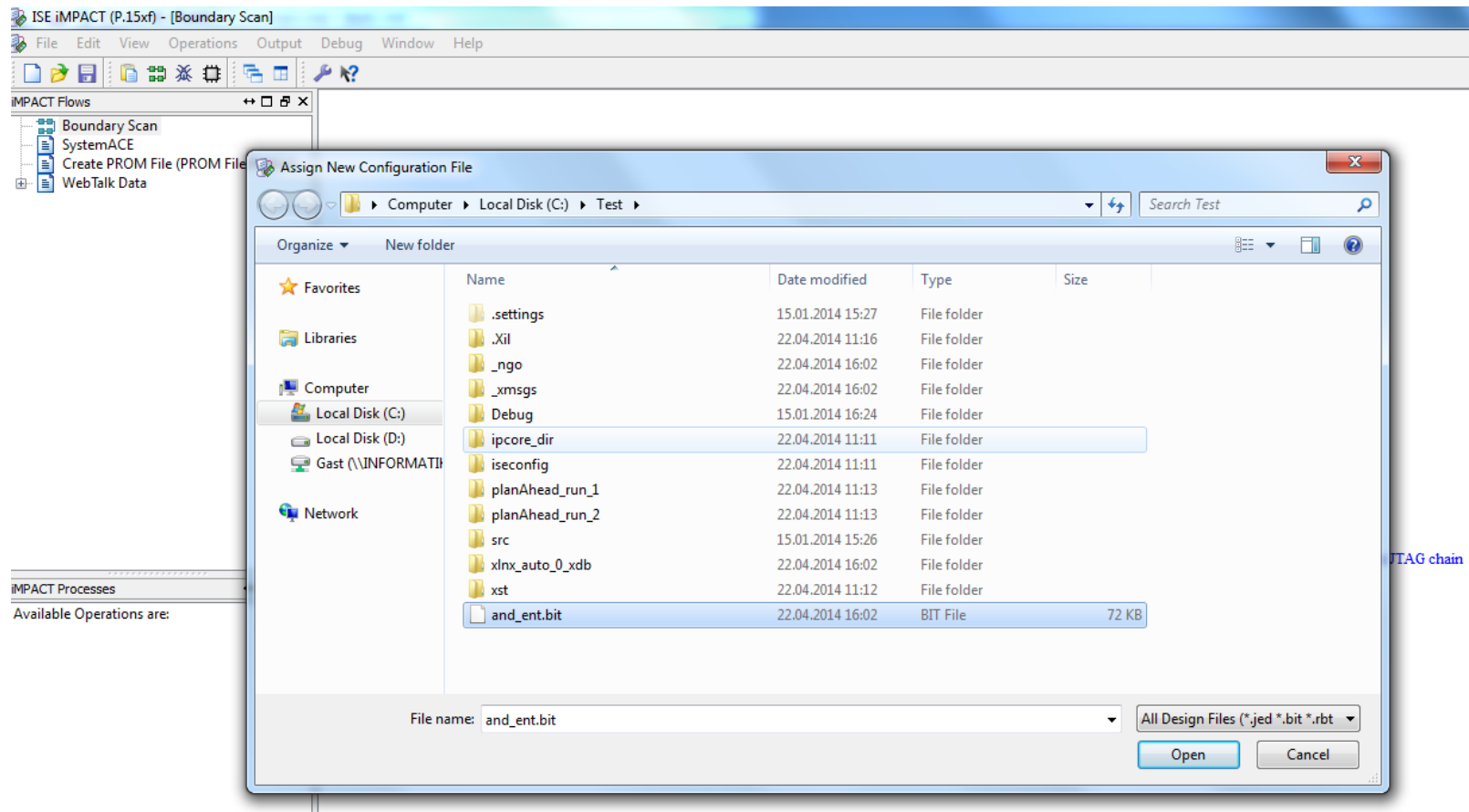
24b. If board is connected and Switched On then you will have the following output



Identify Succeeded



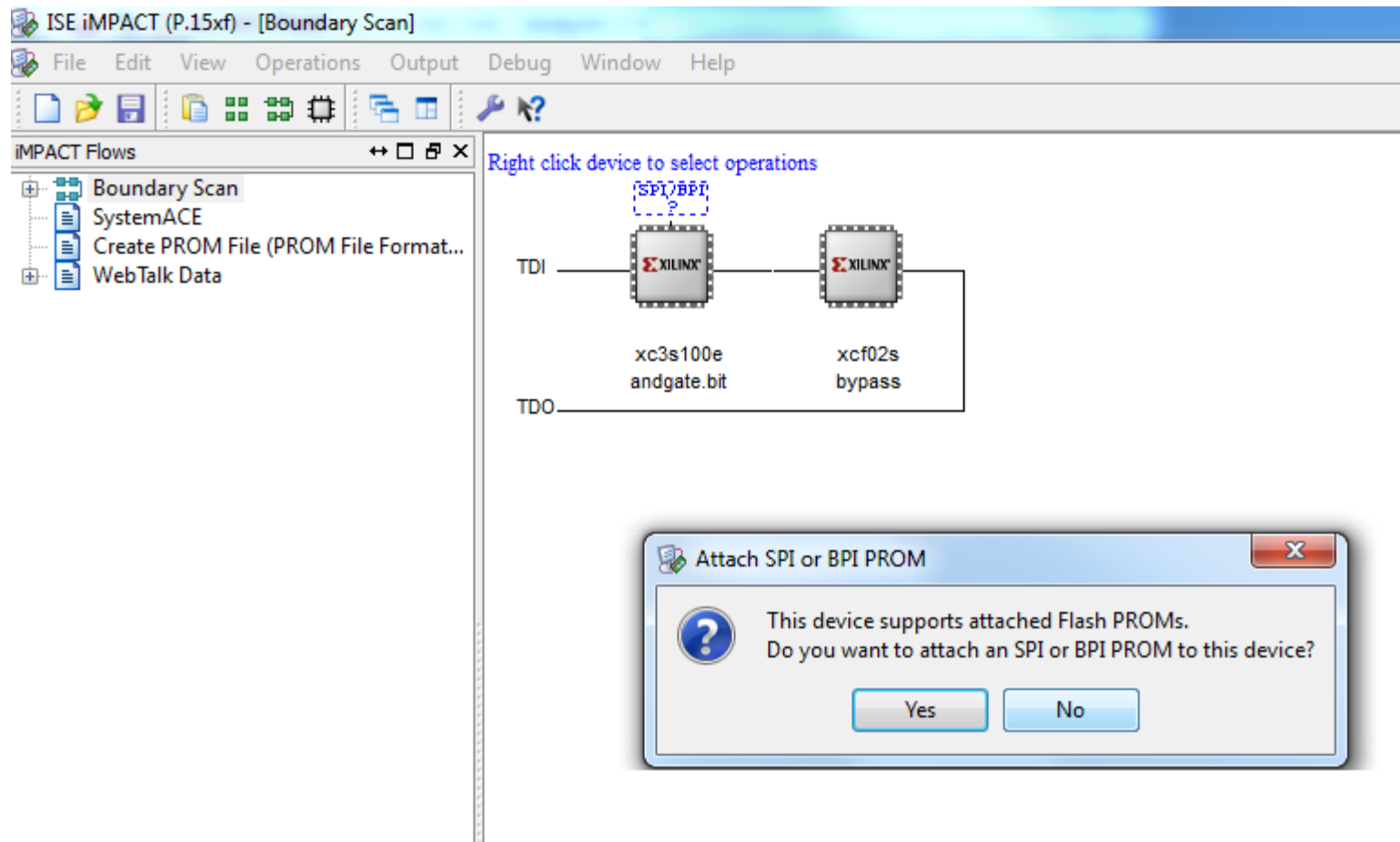
25. Now select the bit file and click open.



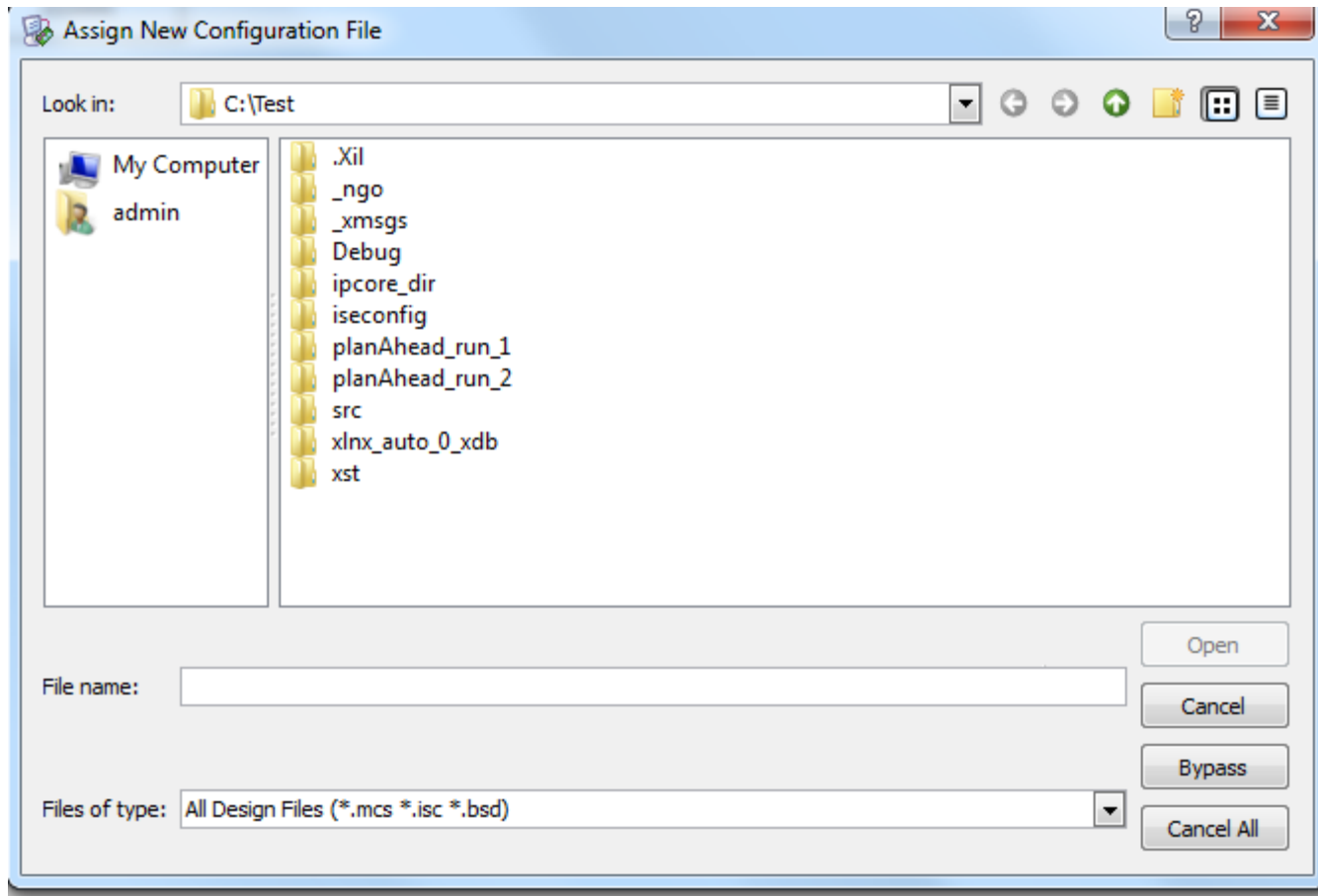
ITAG chain



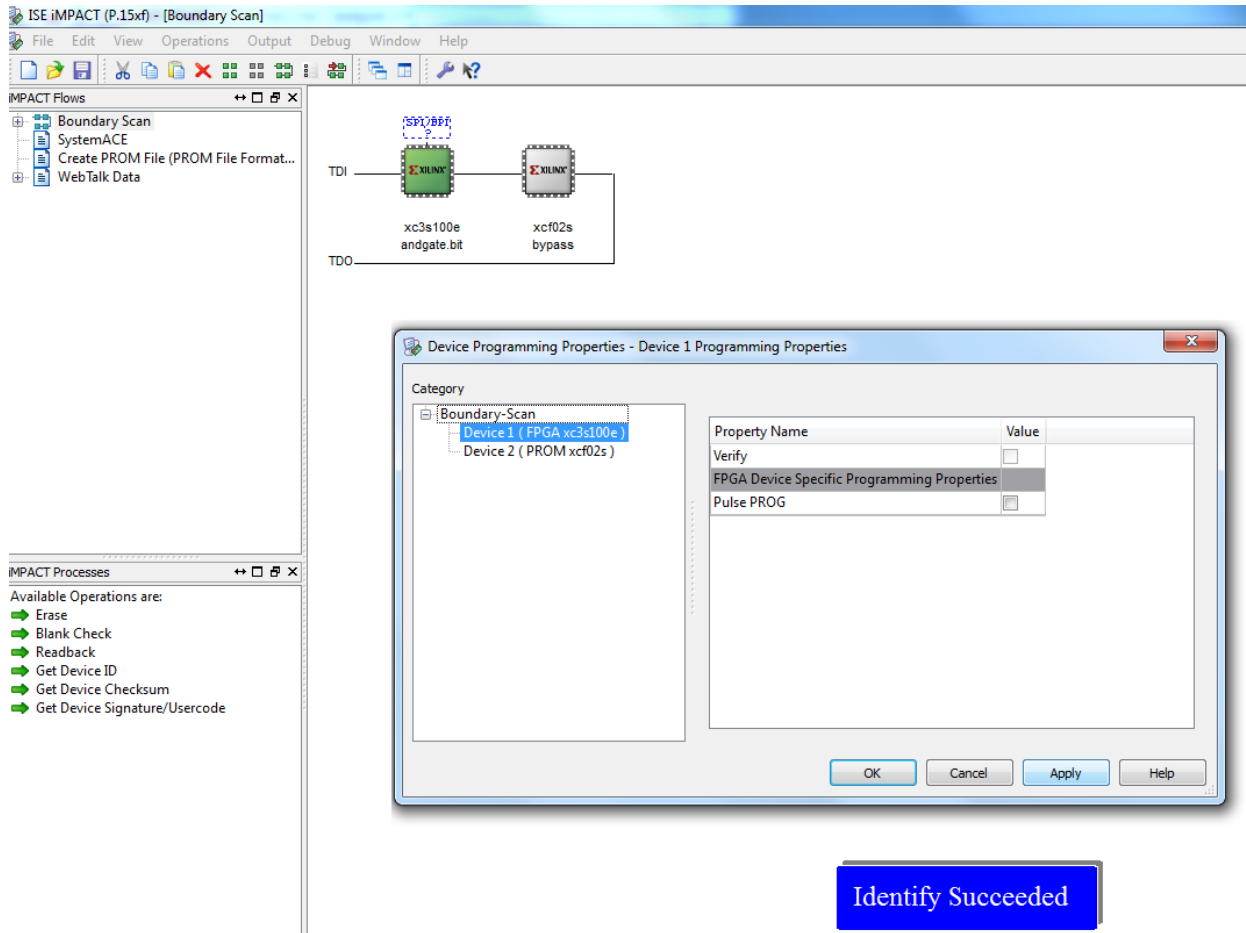
We are not using PROMS so click No.



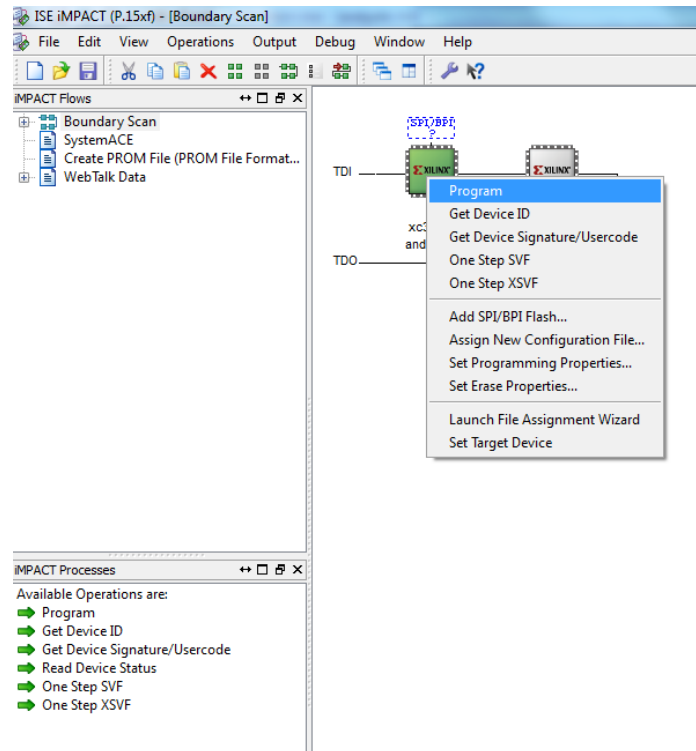
26a. Click Bypass



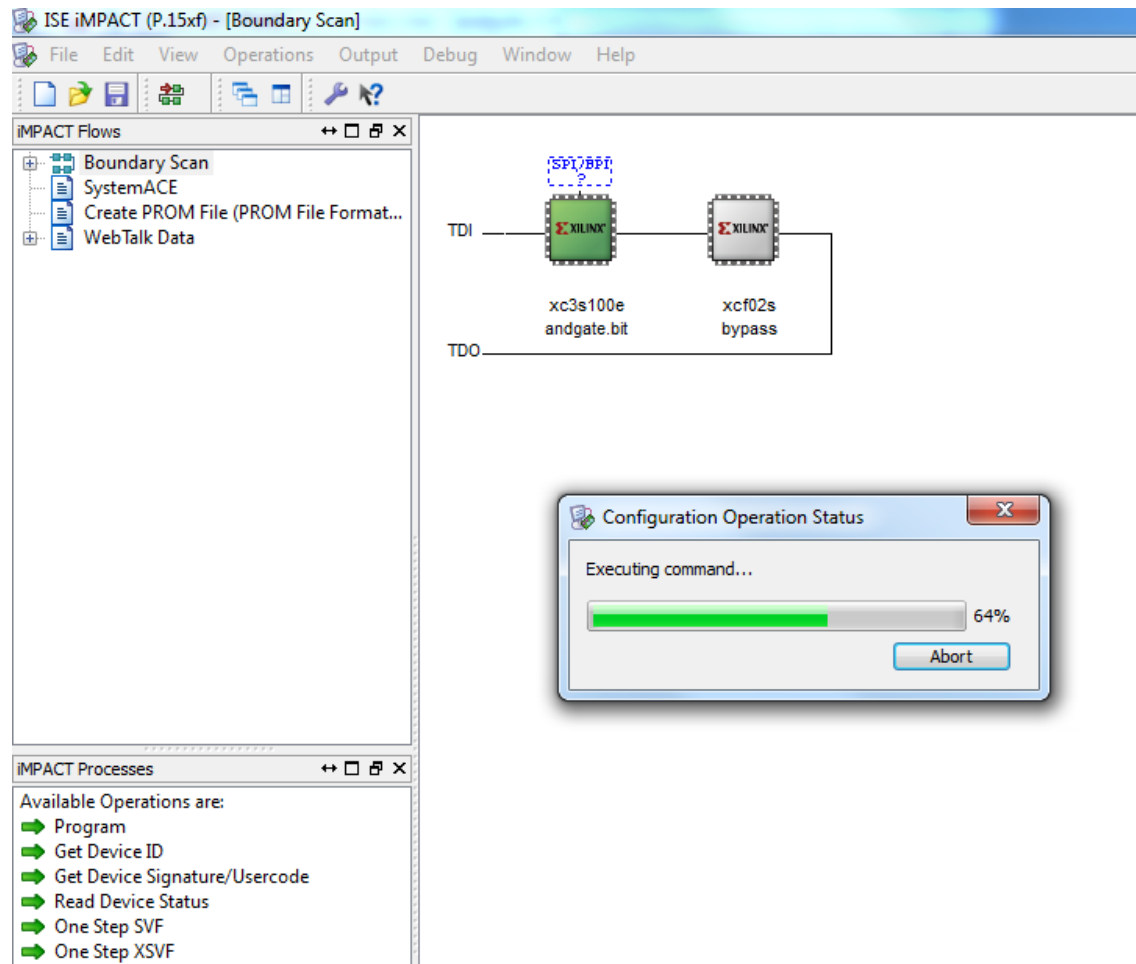
26b.Click Ok .



27a. Right Click on the device, It will be turned Green, Now select Program.



27b.You will see a red light flashing and status bar.



28. If the program is successfully downloaded on the board then the output will appear as shown below.

